



Programowanie na Autopilocie

[Słowem wstępu](#)

[Zdarza Ci się robić coś dla zajawki?](#)

[Twoje otoczenie](#)

[Ludzie którymi się otaczasz](#)

[Źródła wiedzy, po które sięgasz](#)

[Narzędzia z jakich korzystasz](#)

[Twój odpoczynek](#)

[Twoje umiejętności to więcej niż praca](#)

[Konfiguracja systemu](#)

[Konfiguracja sprzętowa](#)

[Tryb stacjonarny](#)

[Ustawienia systemu](#)

[CLI i Skrypty Shell](#)

[Zaprogramuj swojego Maca!](#)

[Wykorzystaj sprytne narzędzia](#)

[Launcher](#)

[Własne polecenia](#)

[Snippety](#)

[API Raycast](#)

[Keyboard Maestro](#)

[Możliwości Keyboard Maestro](#)

[Rozszerzenie możliwości trackpada](#)

[Szybka nawigacja po elementach menu](#)

[Szybkie otwieranie plików](#)

[Szybkie przenoszenie plików](#)

[X-Callback-URL](#)

[Hyper Key](#)

[Praca z tekstem na poziomie](#)

[Automatyczna transformacja tekstu](#)

[JavaScript jest wszędzie](#)

[Własne rozszerzenia i ogromne możliwości](#)

[Twoje mikroserwisy](#)

[VPS](#)

[Połączenie Keyboard Maestro z Twoim API](#)

[Make.com](#)

[Webhooki](#)

[Zapytania HTTP](#)

[Własne moduły Make!](#)

[Twój własny asystent](#)

[Pomysły na automatyzacje](#)

[Kilka inspiracji dla Ciebie](#)

[Twoja Armia Robotów](#)

Słowem wstępu

Nie wiem czy masz tego pełną świadomość, ale programowanie może wyglądać zupełnie inaczej, niż powszechnie się uważa. Z jakiegoś powodu wykorzystujemy nasze umiejętności niemal wyłącznie do realizowania projektów w pracy i niekiedy projektów pobocznych, które czasem rozwijają się, przeradzając w biznes.

Niezależnie od tego jaką ścieżkę w programowaniu wybierzesz, schemat jest taki sam: nauka, praca, życie prywatne i zainteresowania. Balansowanie pomiędzy tymi obszarami jest możliwe ale nie zawsze łatwe. Tym bardziej, że **programowanie zmienia się szybko**. Bardzo szybko.

Jeżeli obserwujesz mnie w mediach społecznościowych lub przynajmniej kojarzysz projekty, które rozwijam, to wiesz że dzieje się u mnie całkiem sporo. W dodatku jest cały świat mojego życia prywatnego, który zwykle zachowuję wyłącznie dla siebie i osób, które go tworzą.

W tym wszystkim, niemal każdego dnia czerpię mnóstwo satysfakcji z tego, co robię. Pomimo rozwoju w wielu obszarach szeroko pojętej technologii i biznesu, jestem na bieżąco z praktycznie wszystkimi istotnymi tematami. Rozwijam firmy, programuję, nagrywam filmy, kursy, podcasty, piszę artykuły, newslettery i rozwijam społeczność oraz kilka pobocznych projektów. Wydaje się, że to dużo jak na jednego człowieka, a przynajmniej tak twierdzą ludzie, którzy obserwują mnie z boku. Dla tych, którzy znają mnie bliżej, przestaje być to takie nadzwyczajne. W chwili gdy mają okazję obserwować **jak pracuję**, wszystko staje się jasne.

Ten e-book jest właśnie o tym. **O tym, aby podzielić się z Tobą tym, jak pracuję**. Zebrałem w jednym miejscu najważniejsze, moim zdaniem, techniki pracy, które możesz po prostu zabrać dla siebie. Po jego przeczytaniu, nie tylko będziesz w stanie

zrozumieć jak to możliwe, że robię to, co robię. **Przekonasz się, że fakt, że potrafisz programować, to daje Ci coś więcej niż Ci się wydaje. Mało tego! Dowiesz się co to może dla Ciebie oznaczać.**

Wszystko co za chwilę przeczytasz, wprost wyciągam ze swojej codziennej pracy. Niektóre elementy możesz skopiować wprost, inne musisz dostosować do siebie (i podpowiem Ci jak) a jeszcze inne mogą z różnych powodów okazać się dla Ciebie mało przydatne, ale ostatecznie możesz potraktować je jako inspirację. **Ponieważ w tym e-booku nie chodzi tylko o to, aby zabrać moje techniki, lecz aby wyrobić sobie styl myślenia, który pozwoli Ci rozwijać Twoje własne.**

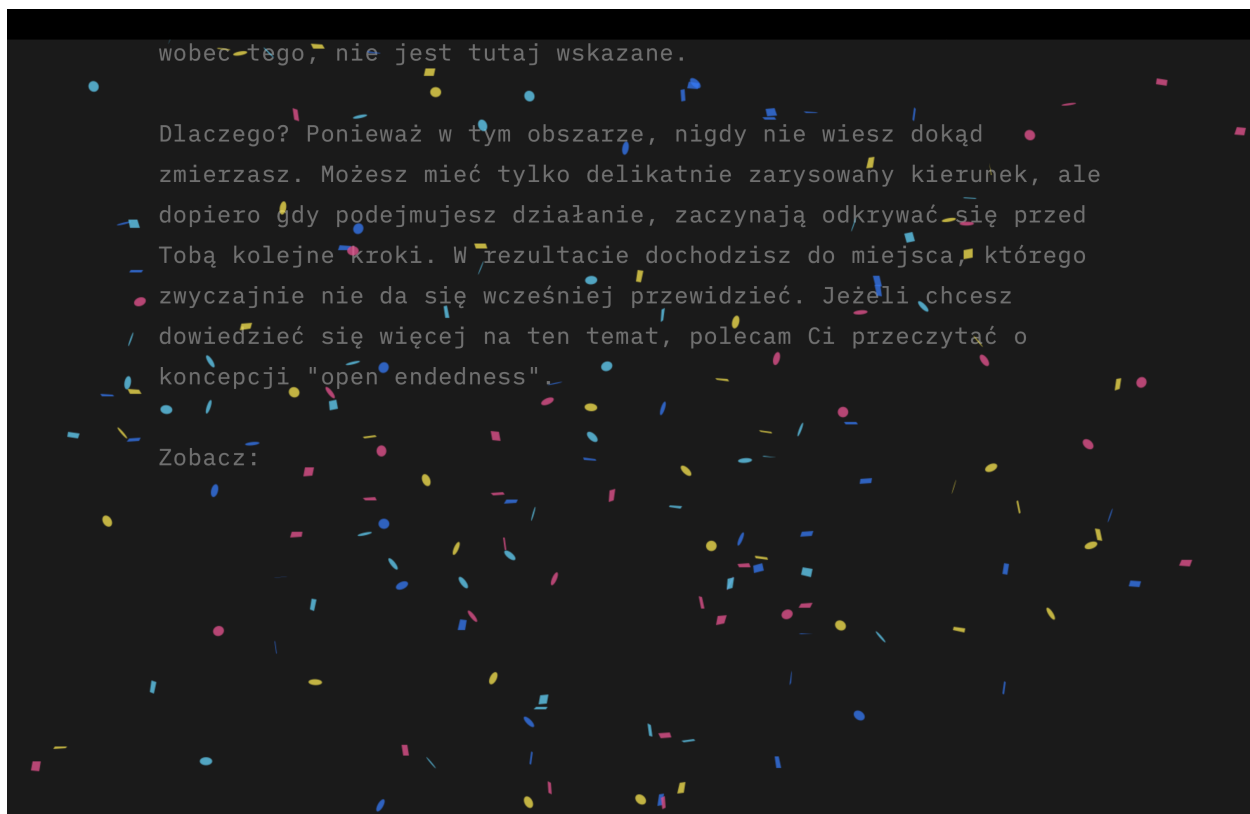
Zdarza Ci się robić coś dla zajawki?

Odkąd pamiętam, bardzo istotnym elementem mojego funkcjonowania oraz procesu nauki, było tworzenie różnych rzeczy wyłącznie dla zajawki. Pozornie może wydawać się to całkowitą stratą czasu, ale właśnie robiąc takie rzeczy, wykorzystuję w praktyce koncepcję, która charakteryzuje ... między innymi "naukowców od rakiet". Wiele o niej możesz przeczytać w książce "Think Like a Rocket Scientist", którą bardzo Ci polecam.

Piszę o tym, ponieważ wszystko o czym powiem Ci za chwilę, rozwijałem właśnie będąc w trybie "zajawki", "zabawy", "ciekawości", "eksperymentu" i "eksploracji". Jest to bardzo istotne, aby nie traktować tego, jak realizację kolejnego zadania w pracy. Powiedziałbym, że nawet zbyt poważne podejście wobec tego, nie jest tutaj wskazane.

Dlaczego? **Ponieważ w tym obszarze, nigdy nie wiesz dokąd zmierzasz.** Możesz mieć tylko delikatnie zarysowany kierunek, ale dopiero gdy podejmujesz działanie, zaczynają odkrywać się przed Tobą kolejne kroki. **W rezultacie dochodzisz do miejsca, którego zwyczajnie nie da się wcześniej przewidzieć.** Jeżeli chcesz dowiedzieć się więcej na ten temat, **polecam Ci przeczytać o koncepcji "open endedness".**

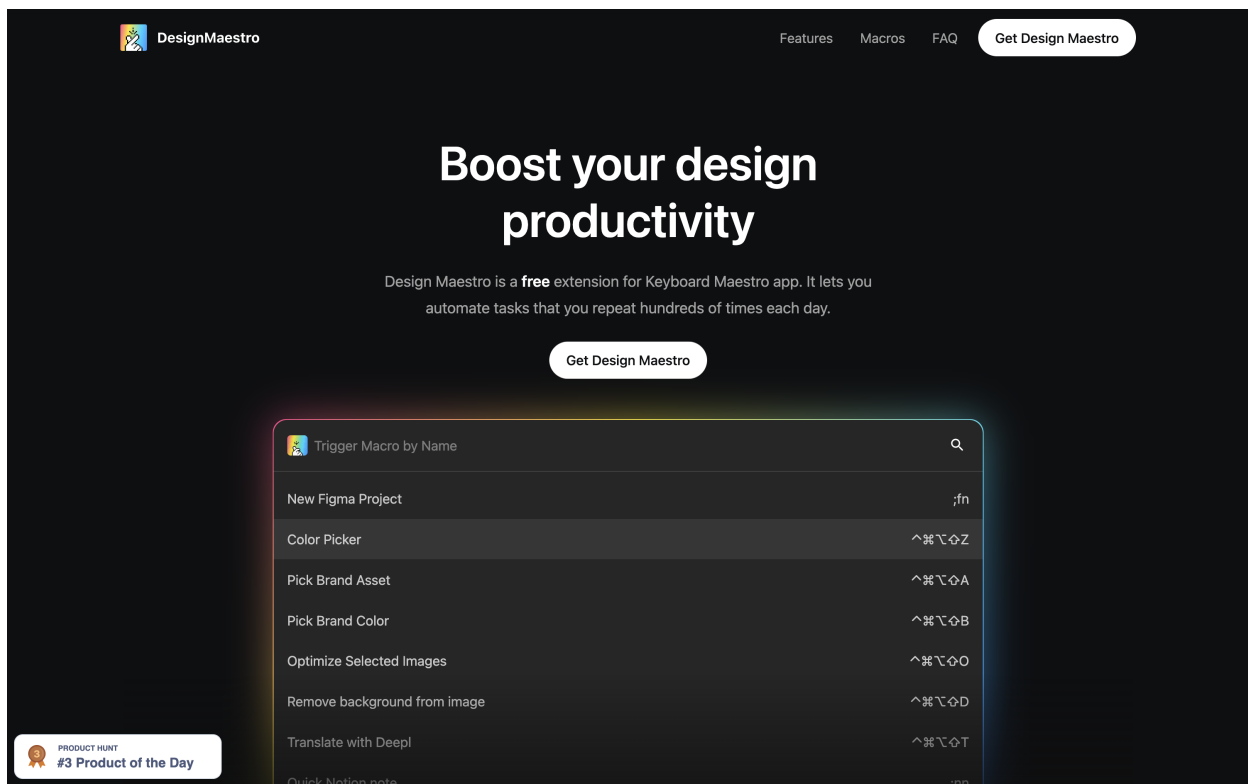
Spójrz na to:



W jednej z wolnych chwil stworzyłem bardzo prosty skrypt, wykorzystujący bibliotekę [canvas-confetti](#). Jest on o tyle ciekawy, że uruchamia się w dowolnym miejscu mojego systemu operacyjnego. W dodatku jest podłączony do skrótu klawiszowego.

Coś takiego pozornie jest całkowicie bezużyteczne, tym bardziej że identyczna funkcja dostępna jest w jednym z programów o których powiem Ci później.

Okazuje się jednak, że w moim przypadku, ten jeden skrypt doprowadził do zrealizowania projektu <https://www.designmaestro.io>, który **zdołał trzecie miejsce na Product Hunt!** Design Maestro istnieje teraz w takiej formie, ponieważ projektując skrypt z Confetti odkryłem prosty sposób projektowania interfejsów, które dziś wykorzystuję też do tworzenia różnego rodzaju makr, które oszczędzają mi dziesiątki godzin pracy każdego miesiąca.



O szczegółach opowiem Ci później, jednak teraz powyższy przykład jasno oddaje to, co chcę Ci tutaj przekazać. Stworzenie sobie odpowiedniej przestrzeni na realizowanie nawet pozornie mało istotnych projektów, pozwala Ci na odkrywanie nowych możliwości. Nigdy nie wiesz która z nich znacząco przełoży się na Twoje codzienne funkcjonowanie oraz pracę. A niekiedy doprowadzi Cię do swojego własnego projektu, który tak jak w moim przypadku, zdobył serca ludzi na całym Świecie.

Ta przestrzeń do eksploracji oraz specyficzny styl myślenia to pierwszy element do tego, aby w pełni wykorzystać wiedzę i techniki, jakie znajdziesz w tym e-booku. Jednocześnie to tylko jeden z elementów, które potrzebujesz zbudować i zacząć aktywnie wykorzystywać.

Twoje otoczenie

Przełączanie się w tryb "dziecięcej ciekawości" i eksploracji, niemożliwe jest w momencie, gdy Twoje otoczenie temu nie sprzyja. Dlatego pierwszym krokiem, który potrzebujesz podjąć w tym kierunku, jest **dostosowanie tego, co Cię otacza, do tego, co chcesz robić**.

Jest to **najważniejsza aktywność** jaką podejmuję. Momentami wręcz brutalne przystosowanie mojego otoczenia, do realizacji moich celów, jest dla mnie priorytetem. Wynika to z faktu, że gdy wszystko co wokół, ułożone jest pod to, co chcę zrobić, trudno jest mi tego nie robić.

Mówiąc odwrotnie - jeżeli nie stworzysz sobie przestrzeni i warunków do tego, aby eksplorować Twoje umiejętności oraz odkrywać nowe możliwości, trudno Ci będzie to robić. Albo mówiąc bardziej dosłownie - **nie będziesz tego robić**.

No bo wyobraź sobie scenariusz, w którym chcesz stworzyć swój side-project lub przetestować jakieś nowe narzędzia o którym ktoś Ci ostatnio powiedział. Jeżeli cokolwiek Ci w tym przeszkadza, trudno będzie Ci to zrobić. Czasem są to sprawy zawodowe, innym razem prywatne a jeszcze innym są to rozpraszacze, które w danej chwili skutecznie przykuwają Twoją uwagę.

Każdego dnia każdy z nas ma do wykorzystania określoną ilość energii. Nie przepalaj jej na "siłowanie" się ze swoim otoczeniem, lecz **całkowicie przekieruj na działanie, które jest dla Ciebie właściwe**.

Jak to dokładnie zrobić? Ja mam kilka sposobów, które rozwijałem przez ostatnie lata. Domyślam się, że niektóre będą wspólne dla Ciebie a inne wręcz przeciwnie.

Ludzie którymi się otaczasz

Pierwszym i najważniejszym elementem Twojego otoczenia są ludzie, którymi się otaczasz i z którymi spędzasz najwięcej czasu. Powszechnie mówi się, że 5-7 najbliższych nam osób ma na nas największy wpływ. **Nie zgadzam się z tym**. Według mnie mają one na nas **całkowity wpływ**. Wręcz dostosowujemy się do nich, ich stylu myślenia, energii i światopoglądu.

Co więcej, wiele wskazuje na to, że nie są to wyłącznie moje obserwacje, ale nawet naukowe stanowisko. Mówią o tym m.in. Dr Andrew Huberman oraz Marek Kaczmarzyk. Według nich, człowiek jest wręcz zaprojektowany właśnie w tym celu - aby synchronizować się z najbliższym otoczeniem, ze szczególnym uwzględnieniem osób, które się w nim znajdują.

W praktyce dla mnie oznacza to nic innego, jak dbanie o to aby spędzać czas z ludźmi, którzy **"zrobili to, co ja chcę osiągnąć i mam z nimi coś wspólnego"**. Muszę przyznać, że działa to fenomenalnie! Jednym z najlepszych przykładów tej zasady, jest moja niemal dziesięcioletnia znajomość z człowiekiem, który przez pierwsze lata był

moim technologicznym mentorem. On wówczas jako bardzo doświadczony CTO, dzielił się swoim doświadczeniem, gdy sam byłem na początku tej drogi.

Jeżeli nie wiesz gdzie możesz spotkać takie osoby, w moim przypadku bardzo często sięgam po społeczności i ludzi których widzę w Internecie. Nawet <https://ahoy.so> stworzyliśmy po to, aby móc otaczać się ludźmi, którzy podobnie jak my, całą swoją postawą i aktywnościami, uosabiają koncepcję "tech outlierów". Nie rzadko takie znajomości zaczynają się bardzo przypadkowo, czasem od jednej wiadomości czy wymienionego komentarza. **Nigdy nie wiadomo.**

Źródła wiedzy, po które sięgasz

Drugim bardzo istotnym elementem Twojego otoczenia są źródła informacji z jakich korzystasz. W Internecie możesz znaleźć wiele wartościowej wiedzy, ale jednocześnie jest tam mnóstwo szumu, który utrudnia dotarcie do niej.

Z tego powodu opracowanie własnego sposobu docierania do możliwie najlepszych źródeł informacji postrzegam jako, najważniejszą umiejętność związaną z poruszaniem się w Internecie. Tym bardziej, że podobnie jak ludzie, informacje które docierają do nas w Internecie również mają wpływ na nasze poglądy, zachowania i emocje.

"Garbage In, Garbage Out" - prosta zasada. Jeżeli otaczasz się niskiej jakości wiedzą, nie spodziewaj się że zamienisz to w coś wartościowego i dobrego dla Ciebie.

Naturalnie na koniec dnia jest to Twoja decyzja, natomiast sam świadomie działam w trybie "**write mostly**". Polega on na tym, że odwracam proporcje korzystania z treści w porównaniu z praktycznym wykorzystaniem wiedzy, którą już posiadam. Połączenie tego z solidnymi źródłami wiedzy, daje sensowny rezultat.

Jeżeli chodzi o wspomniane źródła, wybieram je w oparciu o profile osób, które je tworzą. Np. w temacie neurobiologii sięgam po materiały wspomnianego Andrew Hubermana a w przypadku JavaScriptu czytam wpisy Kyle'a Simpsona czy Erica Elliotta. Dlaczego? Ponieważ według mojej wiedzy, u nich znajdę bardzo wysokiej jakości materiały. Poza tym w przypadku takich osób istnieje wysokie prawdopodobieństwo, że sami korzystają z dobrych źródeł. Dlatego Twitter i obserwowanie osób, które oni obserwują, działa tutaj perfekcyjnie.

W tym miejscu mógłbym podzielić się moimi najlepszymi źródłami pozyskiwania wiedzy, jednak trudno jest mi wybrać te, które mogą pasować do Ciebie. Na początek warto jest zadbać o odpowiedni feed w mediach społecznościowych, newsletterze oraz aplikacji

agregujących (np. Feedly). Bardzo lubię czytać też artykuły dostępne na Medium, ponieważ ich algorytmy sugerujące wartościowe wpisy, działają świetnie.

W późniejszej części e-booka podzielę się z Tobą moimi najlepszymi źródłami wiedzy na temat automatyzacji oraz optymalizacji swojej pracy. Tymczasem spróbuj:

- usunąć subskrypcje newsletterów, które trafiają do Twojej skrzynki ale już ich nie czytasz
- usunąć z obserwowanych profile, które Twoim zdaniem bardziej Ci przeszkadzają niż pomagają
- podejrzeć jakie profile obserwują osoby, które chętnie obserwujesz i również je zaobserwuj
- sprawdź czy osoby które zwróciły Twoją szczególną uwagę prowadzą newsletter bądź bloga i dodaj je do swoich źródeł

Na koniec dnia i tak najważniejsze nie jest to, jaką listę źródeł posiadasz, ale to czy faktycznie wykorzystujesz wiedzę, która jest w nich zawarta.

Narzędzia z jakich korzystasz

Narzędzia to kolejny bardzo istotny element mojego funkcjonowania. Akurat aktualnie jestem już na etapie w którym mam dostęp do high-endowych konfiguracji sprzętowych oraz dowolnych narzędzi, które są dostępne na rynku, włączając w to plany Enterprise i wszystkie możliwe funkcje.

Nie zmienia to jednak faktu, że przez długi czas nie miałem do nich dostępu, lecz nadal opierałem się o **możliwie najlepsze narzędzia**, jakie były w moim aktualnym zasięgu.

Nie bez powodu moje aktualne miejsce pracy wygląda właśnie w ten sposób:



Jest to miejsce w którym wręcz uwielbiam przebywać. Porządek który tutaj panuje oraz jakość sprzętu z którym mam do czynienia, ułatwiają mi skupienie oraz przełączanie się

we wspomniany tryb "eksploracji" i "zabawy".

Naturalnie nie chodzi mi tutaj o to aby w Twoim przypadku również wydawać dziesiątki tysięcy na sprzęt, lecz o to aby dbać o to, jakie narzędzia wykorzystujesz w swojej pracy. Sam podchodzę do tego tak, że skoro w tym miejscu spędzam dużą część mojego dnia, to chcę aby było to dobre miejsce.

Podobnie jest ze wszystkimi aplikacjami i usługami z których korzystam. Wielokrotnie ktoś pyta mnie **"ile ja na to wszystko wydaję?"** podczas gdy pytanie powinno brzmieć **"ile ja na tym wszystkim oszczędzam"** albo **"ile to wszystko mi daje?"**.

Narzędzia to kolejny istotny element Twojego otoczenia. Zadbaj o to, aby ułatwiały Ci funkcjonowanie i tworzyły przestrzeń, w której możesz swobodnie tworzyć, co tylko przyjdzie Ci do głowy.

Twój odpoczynek

Nie ma cienia wątpliwości, że do efektywnego funkcjonowania potrzebujesz nauczyć się jak skutecznie odpoczywać. Jest to szczególnie istotne nie tylko ze względu na Twój poziom energii, ale przede wszystkim na fakt, że w dłuższej perspektywie czasu, brak wystarczającej ilości (i jakości) odpoczynku, wróć do Ciebie w różnych negatywnych formach. Warto więc już na wczesnym etapie odnaleźć sposoby na to, aby się regenerować.

Poza tym moment w którym odpoczywasz służy nie tylko temu aby ładować baterie. Tutaj ponownie do gry wchodzi nauka, która wydaje się potwierdzać, że aktywności takie jak spacer, jazda rowerem czy zwykła nuda, sprzyjają kreatywności i rozwiązywaniu problemów.

W moim przypadku odpoczynek uwzględnia aktywności takie jak: jazda samochodem, czytanie książek, słuchanie muzyki, rysowanie, spotkania z przyjaciółmi, spacer z psem i odwiedzanie fajnych restauracji. W Twoim przypadku może to być coś innego. Jednak dobrze aby było dla Ciebie jasne co to dokładnie jest i aby było stałym elementem Twojego funkcjonowania. **Jest to tak istotne, że jeżeli czujesz, że na ten moment ten obszar nie jest u Ciebie zaadresowany, zajmij się nim w pierwszej kolejności.**

Greg Mckeown w Esencjaliście napisał:

| In order to have focus, we need to escape to focus.

I taka zasada działa również dla kreatywności. Musisz mieć przestrzeń w której nie robisz nic, aby potem zrobić coś. To prosta zasada, która przez wielu jest całkowicie pomijana.

Twoje umiejętności to więcej niż praca

Spośród wszystkich znanych mi osób, które potrafią programować, dosłownie tylko garstka wykorzystuje tę umiejętność do rzeczy nie związanych z pracą. Jest to dla mnie wręcz wyjątkowo dziwne, że zdolność do "komunikacji z maszynami" wykorzystywana jest tylko po to aby zrealizować planowaną listę zadań w ramach pracy.

Kiedy ostatni raz zdarzyło Ci się stworzyć jakiś skrypt, aplikację czy mały projekt, który służy Ci każdego dnia? Albo przynajmniej umożliwił Ci zrobienie czegoś, co nie jest dostępne dla osób, które nie potrafią programować?

Pójdźmy dalej - czy zdarza Ci się pisać Skrypty Shell aby wykonywać jakieś powtarzalne czynności? Jak często to robisz? Czy wiesz, że w systemie macOS funkcjonuje AppleScript, który pozwala Ci wchodzić w interakcję niemal ze wszystkimi opcjami Twojego komputera i dużej części aplikacji z jakich korzystasz?

A może posiadasz swój własny VPS, na którym funkcjonują jakieś aplikacje stworzone np. w node.js, które pomagają Ci na różne sposoby?

Rozumiesz już o co mi chodzi? Właśnie takie rzeczy chcę Ci nie tylko uświadomić, ale także pokazać w praktyce. Jeżeli chociaż na część z powyższych pytań odpowiadasz twierdząco, to super! Pójdziemy z tym dalej. Jeżeli to wszystko jest dla Ciebie obce, również się nie przejmuj, ponieważ zaczniemy od początku.

Na koniec dnia istotne jest dla mnie tylko to, aby fakt, że "rozmawiasz z maszynami" został przez Ciebie maksymalnie wykorzystany. W ten sposób, wykorzystamy potencjał jaki w sobie masz. A w konsekwencji ... zbudujesz swoją własną Armię Robotów, podobnie jak ja!

Zaczynamy?

To lecimy!

Konfiguracja systemu

Pierwszym i podstawowym elementem Twojego funkcjonowania jest Twój komputer. Spędzasz przy nim dużą część dnia i z tego powodu Twoim priorytetem powinno być

to, aby funkcjonował jak należy. Byłoby idealnie gdyby sprzęt i jego konfiguracja pomagały Ci na każdym kroku, ale **minimalnym wymaganiem jest to, aby Ci nie przeszkadzały.**

Jak wiesz pracuję na Macbooku i dzieje się to nie bez przyczyny. Pomijając już preferencje i sympatię do marki Apple, faktem jest że szczególnie od premiery procesorów M1, jest to sprzęt o którego jakości trudno jest dyskutować. Z tego powodu nie zamierzam Cię przekonywać ale mogę śmiało powiedzieć, że zakup pierwszego Macbooka był dla mnie najbardziej przełomowym wydarzeniem odkąd programuję. Jeżeli dołożymy do tego moment w którym zacząłem odkrywać możliwości tego systemu a momentami nawet przekraczać ich granice, to trudno będzie opisać słowami to, jak bardzo wpłynęło to na moje funkcjonowanie.

Za chwilę przekonasz się, dlaczego tak piszę. Niemal wszystkie aplikacje o których powiem a nawet możliwości które nie są z nimi bezpośrednio związane, dostępne są **wyłącznie w systemie macOS lub nawet ekosystemie Apple.**

Poza tym sam dosłownie na każdym kroku przekonuję się jak wiele zadań, które np. na Windowsie wymagają kilku godzin konfiguracji, na macOS (lub Linux) sprowadzają się do uruchomienia jednego polecenia w terminalu.

Zakładam jednak, że skoro czytasz ten e-book, posiadasz już swojego Maca i wiesz co mam tutaj na myśli.

Konfiguracja sprzętowa

Zacznijmy od tego, że pracuję przez większość czasu w trybie stacjonarnym. Efektywna praca na małym ekranie jest iluzją i oszukiwaniem siebie. Nie twierdzę, że nie jest możliwa. Jednak jeżeli zestawisz ją z komfortowym stanowiskiem i dużym ekranem lub ekranami, nie ma tutaj o czym mówić.

Poza tym samo miejsce pracy - fotel, biurko i wszystkie drobne elementy, które znajdują się w zasięgu Twojego wzroku, wpływają na to jak pracujesz. To wszystko ponownie składa się na Twoje otoczenie, które może Ci sprzyjać lub przeszkadzać. W moim przypadku stawiam na to pierwsze.

Tryb stacjonarny

Różne badania wskazują na to, że praca na dwóch zewnętrznych monitorach zwiększa efektywność. Nie zamierzam z nimi dyskutować i nawet się z nimi zgadzam. Pomimo tego od roku pracuję wyłącznie na jednym ale za to większym ekranie. Wynika to z

faktu, że bardziej istotne jest dla mnie skupienie niż efektywność wynikająca z większej powierzchni roboczej.

W rezultacie zamieniałem swoje 2x monitory 27" na jeden 32" i od tego momentu, nie myślałem aby wracać do starej konfiguracji.

Na moim biurku przez większość dnia poza klawiaturą i trackpadem znajduje się wyłącznie jeden kubek z wodą lub kawą. Dbam o to aby był tam praktycznie 100% porządek przez cały czas. Wszystkie dodatkowe akcesoria i urządzenia (np. słuchawki czy tablet) spędzają tam maksymalnie kilka chwil. **Skupienie i przejrzystość jest dla mnie priorytetem.** Podobnie też mój telefon niemal zawsze znajduje się poza zasięgiem mojej ręki, dzięki czemu nie sięgam po niego nieustannie.

Ciekawostką jest fakt, że nie korzystam z myszki, ale w zamian wykorzystuję zewnętrzny trackpad i wynika to z kilku rzeczy. Przede wszystkim trackpad daje mi możliwości konfigurowania gestów, które działają zarówno w trybie stacjonarnym jak i mobilnym. Potem pokażę Ci co dokładnie mam przez to na myśli, a jest o czym mówić.

Nowość

Gładzik Magic Trackpad – obszar Multi-Touch w czerni

749,00 zł

Kolor - czarny obszar Multi-Touch



 Dostawa:
w magazynie
Bezpłatna dostawa
[Zobacz daty dostawy](#)

Włóż do torby

Nadal się zastanawiasz?
Dodaj ten produkt do listy, żeby łatwo
wrócić do niego później.



Dodatkowym istotnym elementem mojego wyposażenia jest biurko z opcją pracy na stojąco. Nie jestem pewien dlaczego tak się dzieje, ale przełączanie się pomiędzy

trybem siedzącym a stojącym, daje mi wręcz nową pulę energii do wykorzystania. Niektóre z aktywności wykonuję stojąc a inne wyłącznie siedząc wygodnie w fotelu.

Wykorzystywany przeze mnie sprzęt to:

- biurko DeskWise (ale polecam <https://fully.com>)
- monitor Apple Pro Display (ale polecam Apple Studio)
- macbook full-spec z procesorem M1 Max
- Magic Keyboard
- Magic Trackpad
- słuchawki AirPods Max
- fotel Herman Miller

Jak łatwo zauważysz, wszystko co napisałem powyżej traktuję bardzo poważnie. Nie bez powodu wydałem niemal 100 000 zł na sprzęt na którym pracuję i który stanowi moje najbliższe otoczenie. Naturalnie nie oznacza to tego, że Ty też musisz. Ale w ten sposób masz jasny dowód na to, że **wszystko co piszę, nie jest "tylko mówieniem" ale przede wszystkim działaniem.**

W tym wszystkim chodzi wyłącznie o to aby Twoje najbliższe otoczenie **służyło Ci jak najlepiej.** Dla mnie jest to najwyższej jakości sprzęt, dla kogoś innego może to być coś innego. Ważne jest tylko to aby wykorzystywane narzędzia w żaden sposób Cię nie blokowały lub też ewentualnie przeszkadzały tak mało, jak to tylko możliwe.

Dodam jeszcze że wszystkie zmiany w swojej konfiguracji oraz wykorzystywanych przez Ciebie urządzeniach, możesz wprowadzać stopniowo, nawet bardzo małymi krokami. W ten sposób nawet łatwiej dostosujesz to wszystko do swoich potrzeb. Tak w zasadzie jest to tutaj najważniejsze: **dopasowanie do Ciebie.**

Ustawienia systemu

Nie ma zbyt wiele dedykowanych ustawień systemu z jakich korzystam, ale jednocześnie nie jest to domyślna konfiguracja. Zaczniemy od tego, aby stało się dla Ciebie jasne co oferuje Ci Twój komputer oraz urządzenia jakie posiadasz. Mam tu na myśli nie tylko zainstalowane programy, wbudowane funkcje i skróty klawiszowe ale także elementy takie jak tryb Sidecar, Handoff, Universal Control czy możliwości

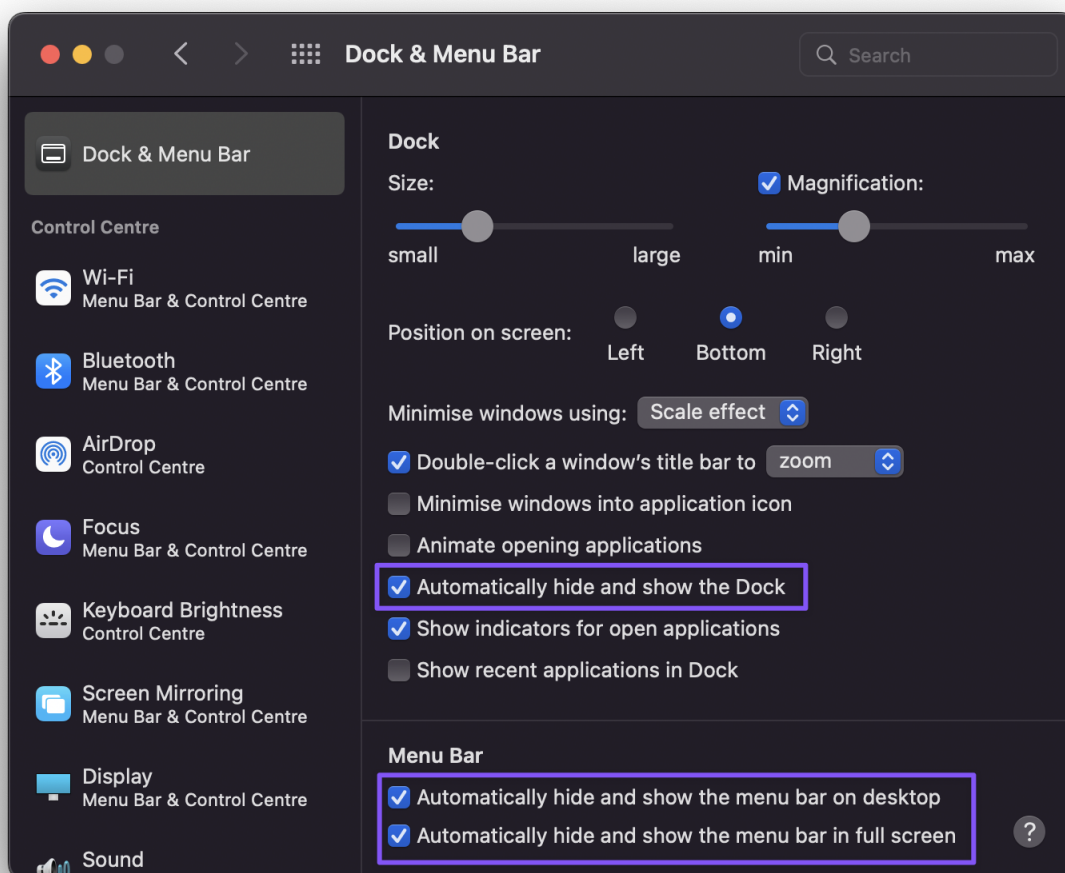
AppleScript. Poza tym jest przynajmniej kilka domyślnych ustawień, które definitywnie warto zmienić już na samym początku.

Ogólne zasady konfiguracji systemu w moim przypadku uwzględniają to, aby utrzymywać go w minimalistycznej formie. Nie posiadam zbyt wielu aplikacji i jeżeli jakaś nie jest już przeze mnie wykorzystywana, zostaje usunięta. Poza tym dbam o to aby niemal wszystko, z czego korzystam często, było dla mnie dostępne najlepiej poprzez skrót klawiszowy lub gest na trackpadzie.

Dodatkowym oczekiwaniem z mojej strony jest to, aby moja konfiguracja działała niezależnie od tego czy pracuję aktualnie w trybie stacjonarnym czy mobilnym. Dzięki temu wszystkie moje nawyki, zawsze są dla mnie dostępne.

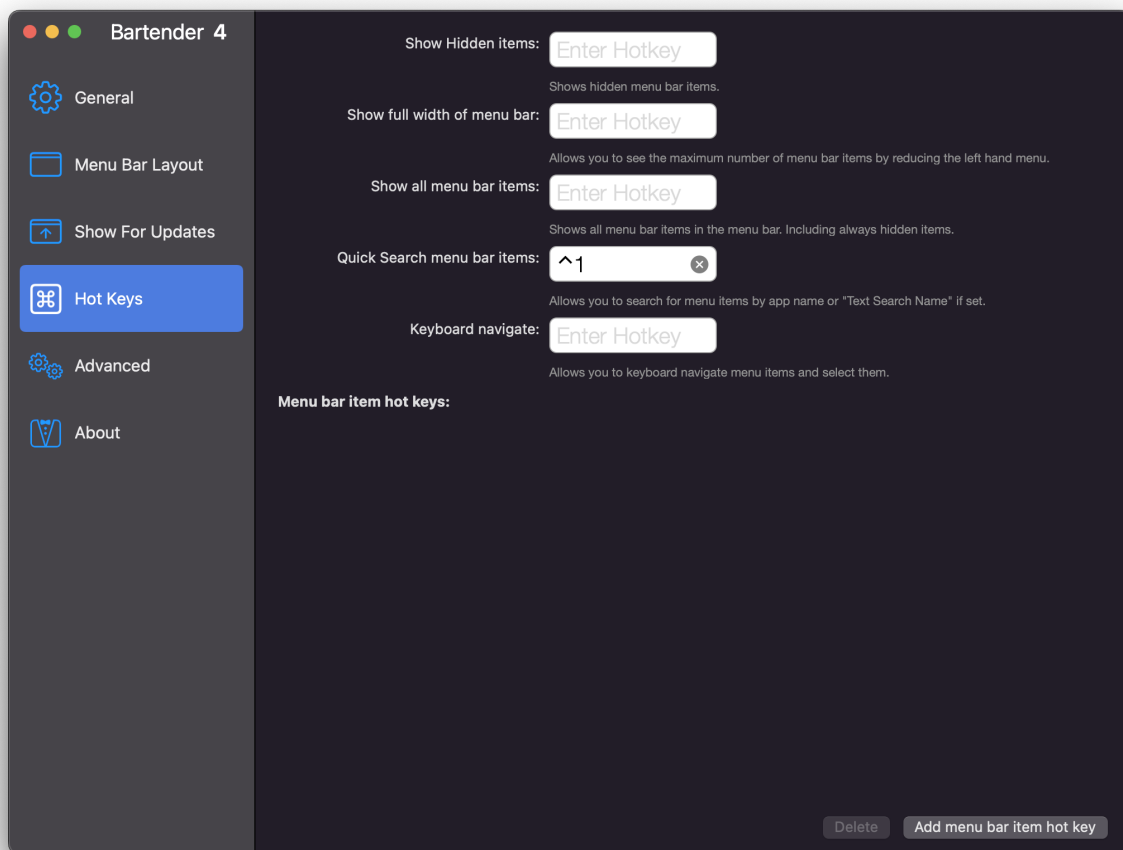
Podstawowe ustawienia

Tutaj nie mam zbyt wiele do powiedzenia, natomiast są dwa szczegóły o które warto zadbać. Pierwszy z nich to ukrywanie menu bar a drugi to ukrycie Docka. W obu przypadkach ustawienie znajdziesz w preferencjach w kategorii **Dock & Menu Bar**

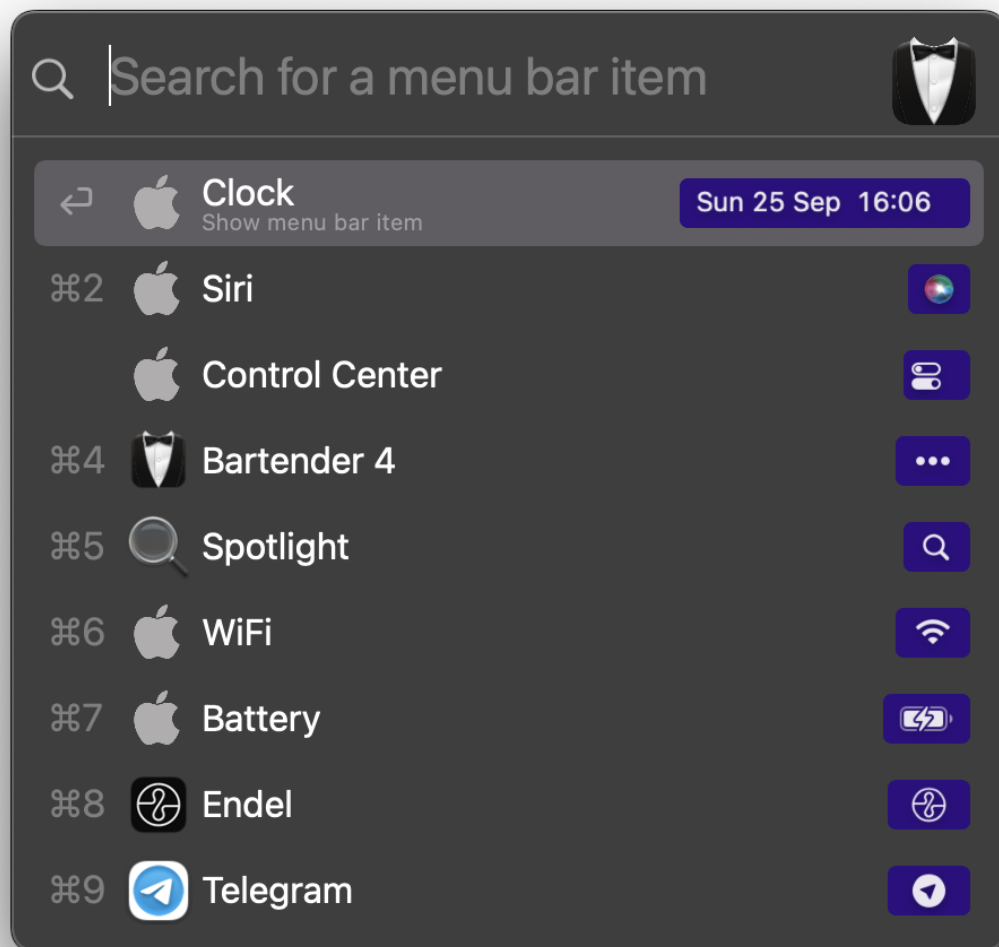


Jest to dobry sposób na to aby oszczędzić trochę miejsca. Dodatkowo bardzo przydatną aplikacją w kontekście paska menu, jest Bartender. Zamiast jednak kupować go oddzielnie, lepiej będzie jeżeli wykupisz pakiet aplikacji Setapp.

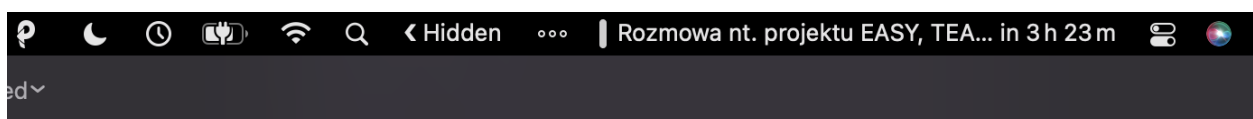
Po instalacji Bartendera przejdź do jego preferencji (kliknij prawym na trzy kropki w menu bar, które pojawiają się po jego uruchomieniu) i w zakładce skrótów klawiszowych przypisz skrót do opcji: `Quick Search menu bar items`.



W ten sposób łatwo dostaniesz się do dowolnego elementu menu:



Muszę przyznać, że w moim przypadku bardzo rzadko sięgam po menu bar. Po pierwsze dlatego, że przez większość czasu jest ukryty a po drugie dlatego, że do większości ustawień dostaję się w inny sposób (opiszę później). Nie zmienia to jednak faktu, że w przypadku niektórych aplikacji konieczne jest wykorzystanie ikony dostępnej w menu bar aby np. ustawić jej preferencje. Wtedy Bartender sprawdza się genialnie, tym bardziej że aplikacje z których korzystasz najczęściej, mogą być zawsze widoczne:



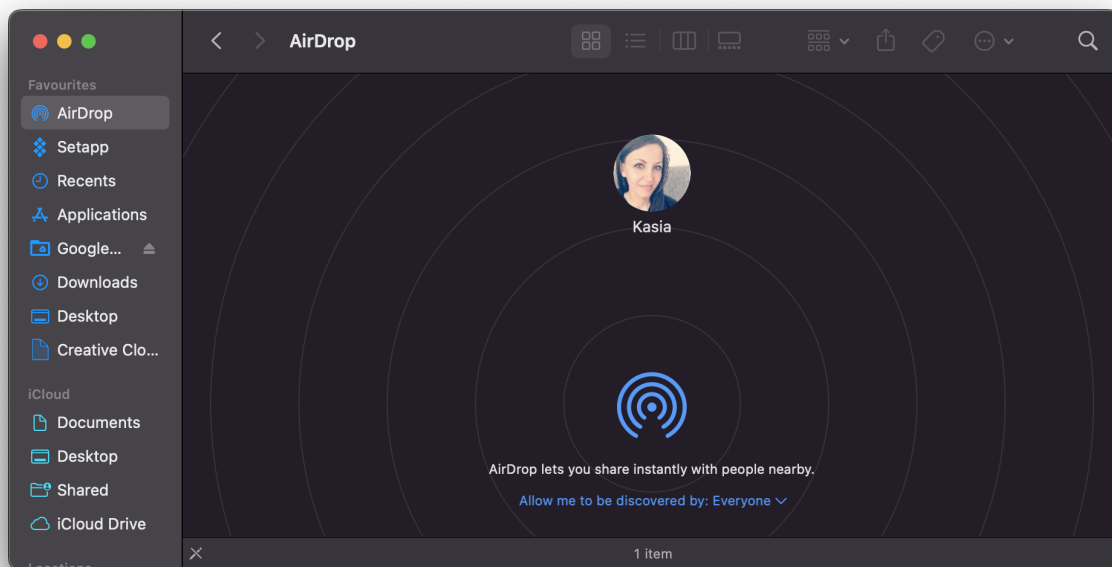
Wystarczy że przytrzymując  przeciągniesz ją na prawą stronę "trzech kropek" Bartendera.

iCloud oraz Apple ID

Jest to raczej standardowa procedura podczas aktywacji systemu operacyjnego. Ważne jest jednak to aby w momencie gdy posiadasz zarówno iPhone jak i Macbooka, zalogować się na obu urządzeniach na to samo Apple ID. W ten sposób zyskasz możliwość pisania SMSów i odbierania połączeń bezpośrednio na swoim Macbooku.

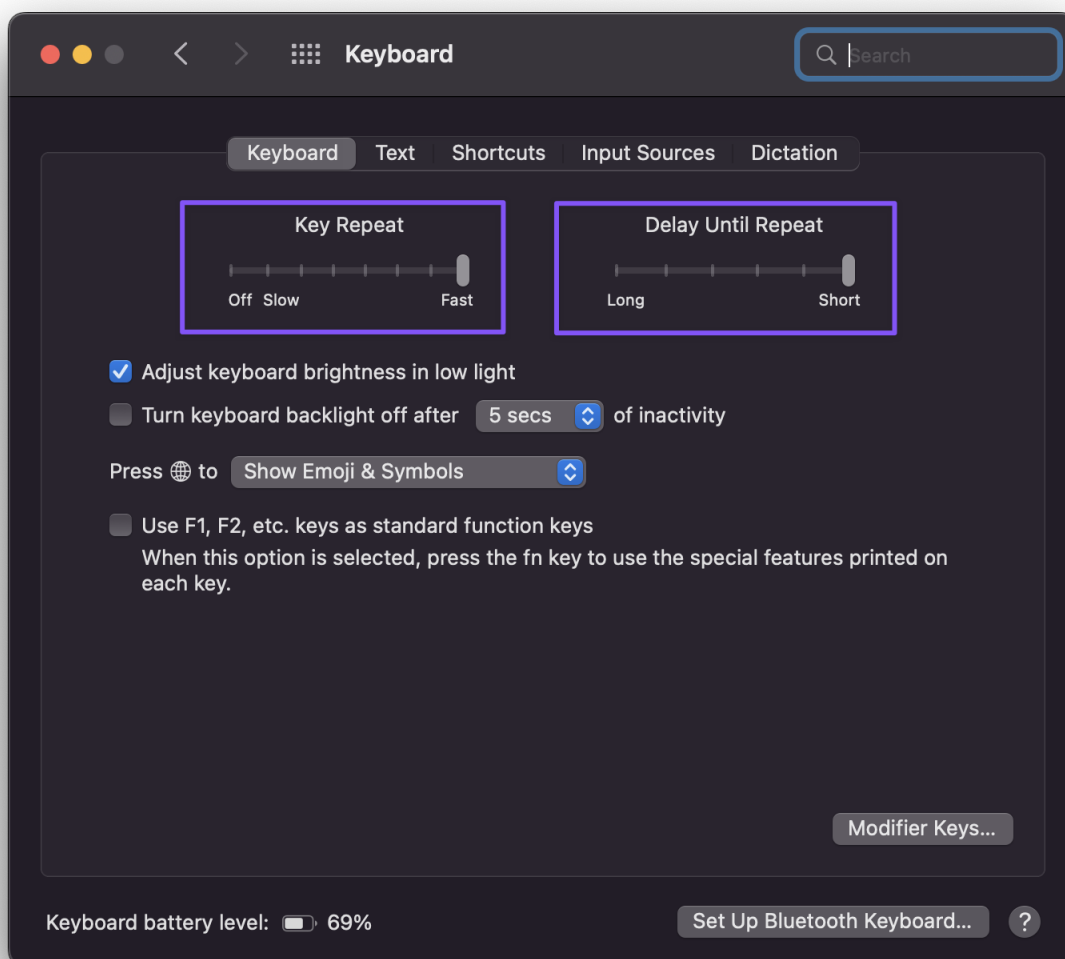
Drugą opcją jest łatwe współdzielenie schowka. Gdy znajdujesz się w tej samej sieci, cokolwiek co skopiujesz na telefonie, pojawi się na komputerze i odwrotnie (to tzw. Handoff).

Poza tym pamiętaj o tym, że pomiędzy urządzeniami Apple możesz łatwo przesyłać pliki z pomocą AirDrop (opcja ta dostępna jest w Finderze w sidebarze po lewej stronie).



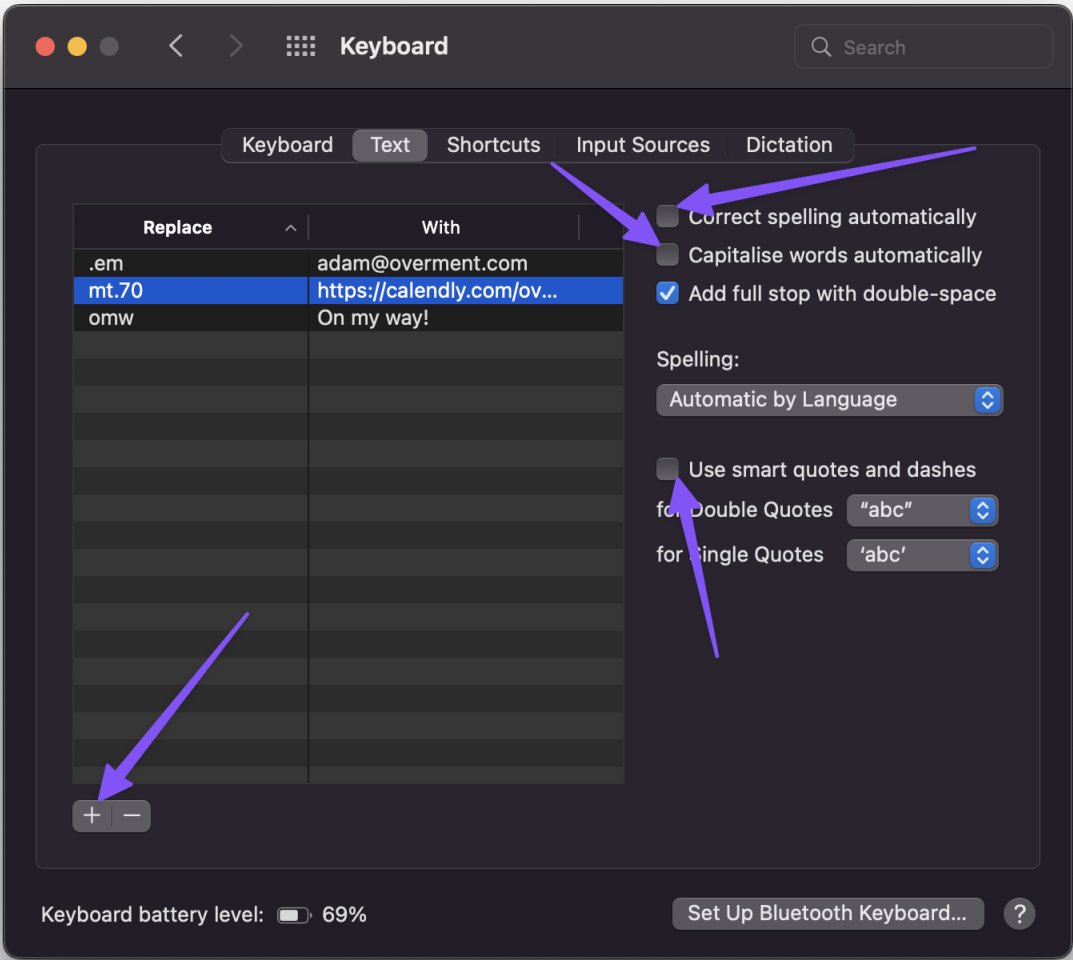
Ustawienia klawiatury

Praca z tekstem jest kluczowa w programowaniu. W ustawieniach klawiatury zmień poniższe dotyczące szybkości powtarzania oraz częstotliwości powtarzania wciśnięcia. W ten prosty sposób przyspiesz to jak szybko porusza się Twój kursor i ta jedna rzecz oszczędzi Ci mnóstwo czasu na przestrzeni kolejnych lat.

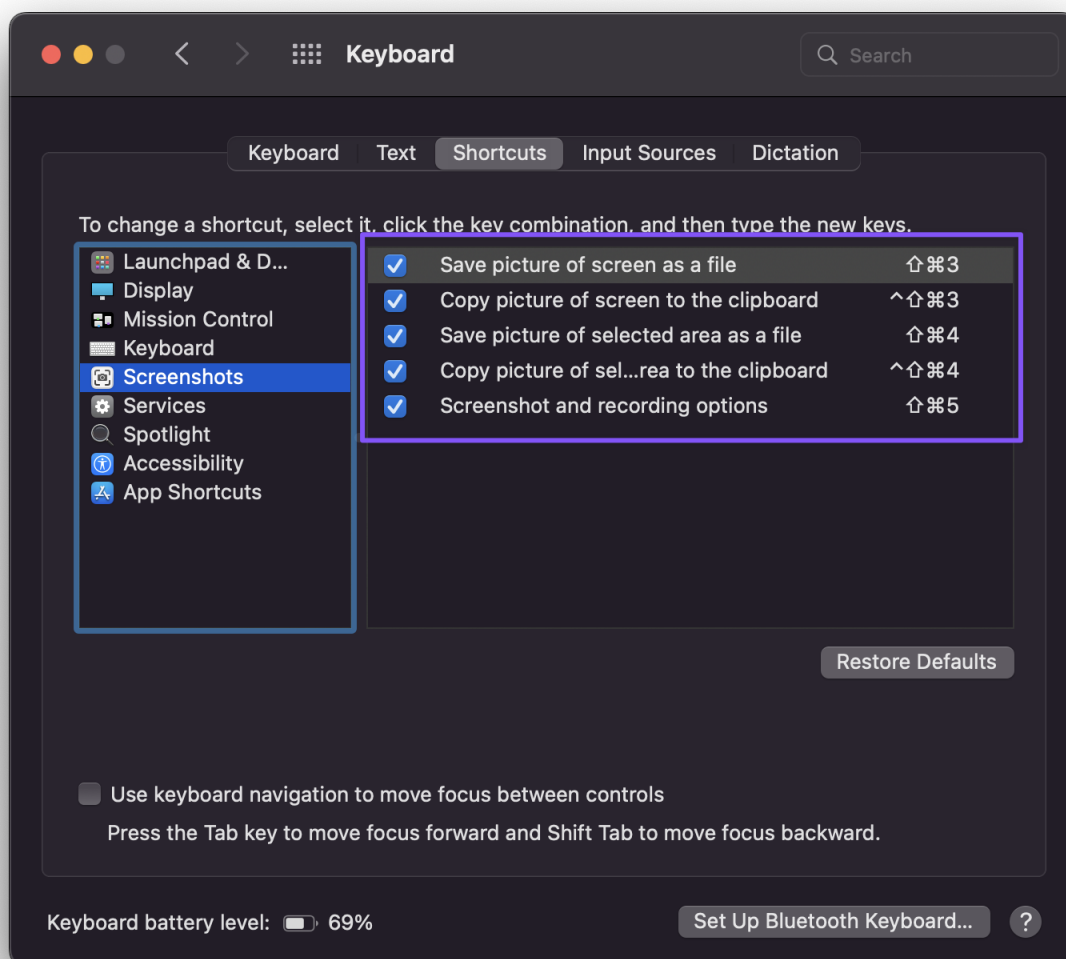


Następnie przejdź do zakładki **Text** i wyłącz opcje autokorekty, powiększania liter, oraz definitywnie używania smart quotes, które na każdym kroku mogą sprawiać Ci problemy w pracy z kodem źródłowym.

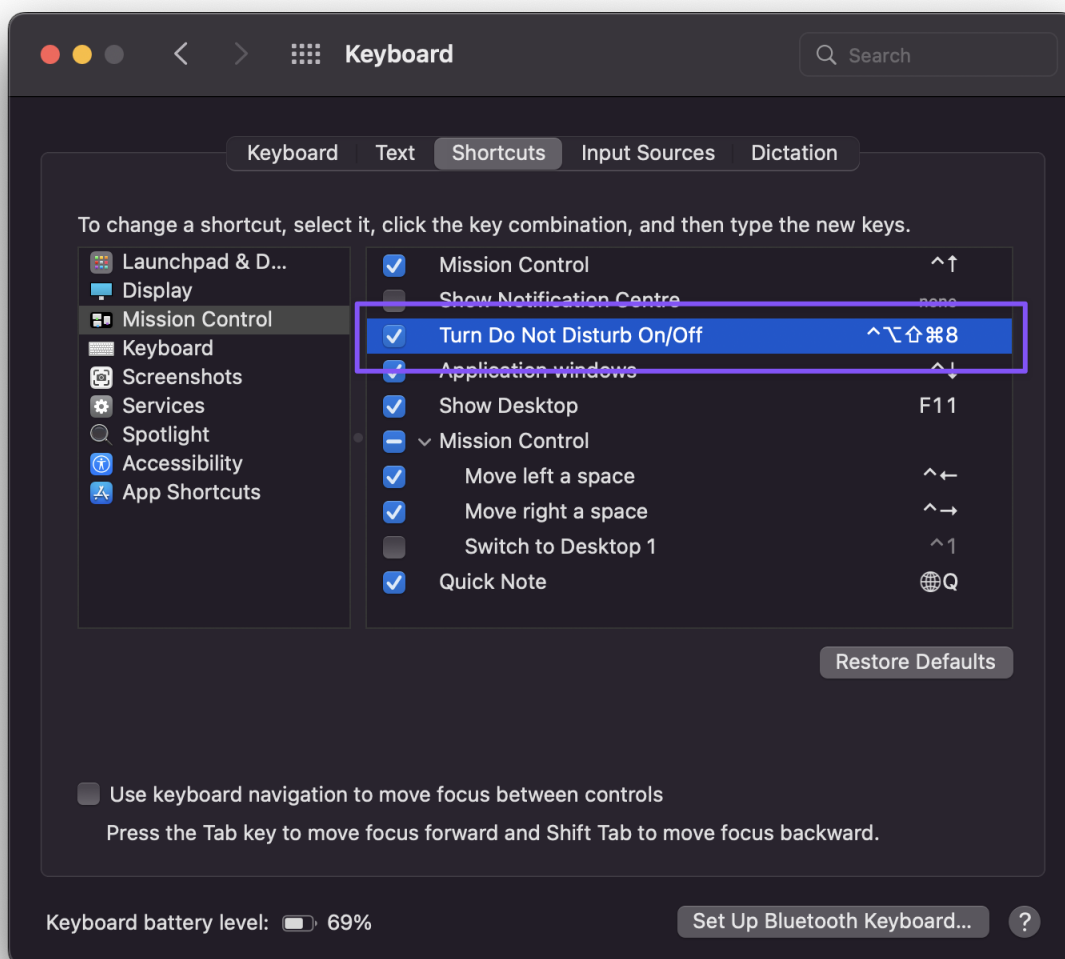
Jako ciekawostkę dodam, że możesz w tym miejscu ustawić Text Expander, dzięki któremu zamiast wpisywać np. swój adres e-mail w całości, wystarczy że wpiszesz krótką frazę, np. .em, która zostanie zamieniona na podany przez Ciebie tekst. Opcja ta działa również na iPhone i iPadzie.



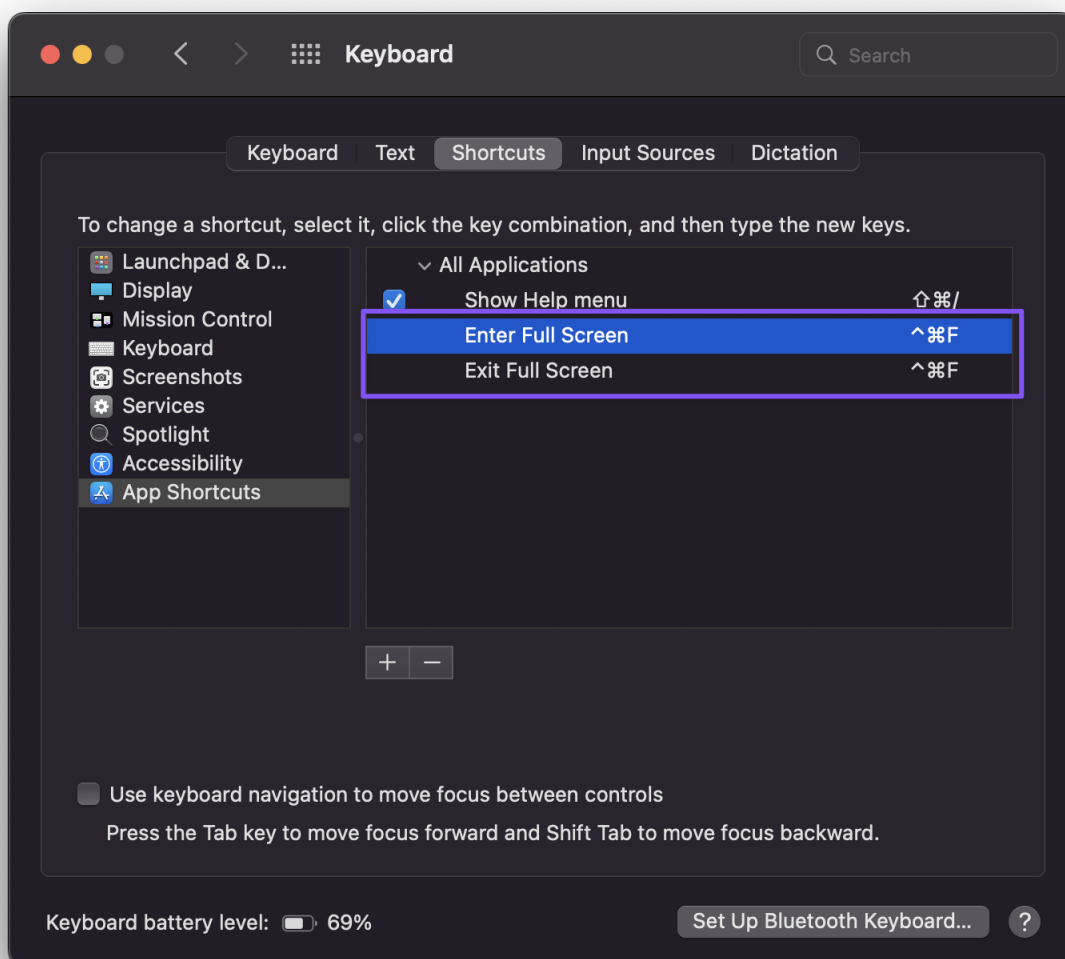
Teraz rzuć okiem jeszcze do zakładki Shortcuts, gdzie znajdziesz różne ciekawe skróty klawiszowe, które funkcjonują w systemie macOS lub mogą zostać tam skonfigurowane. Najbardziej przydatne dla mnie dotyczą zrzutów ekranu:



oraz trybu DND:



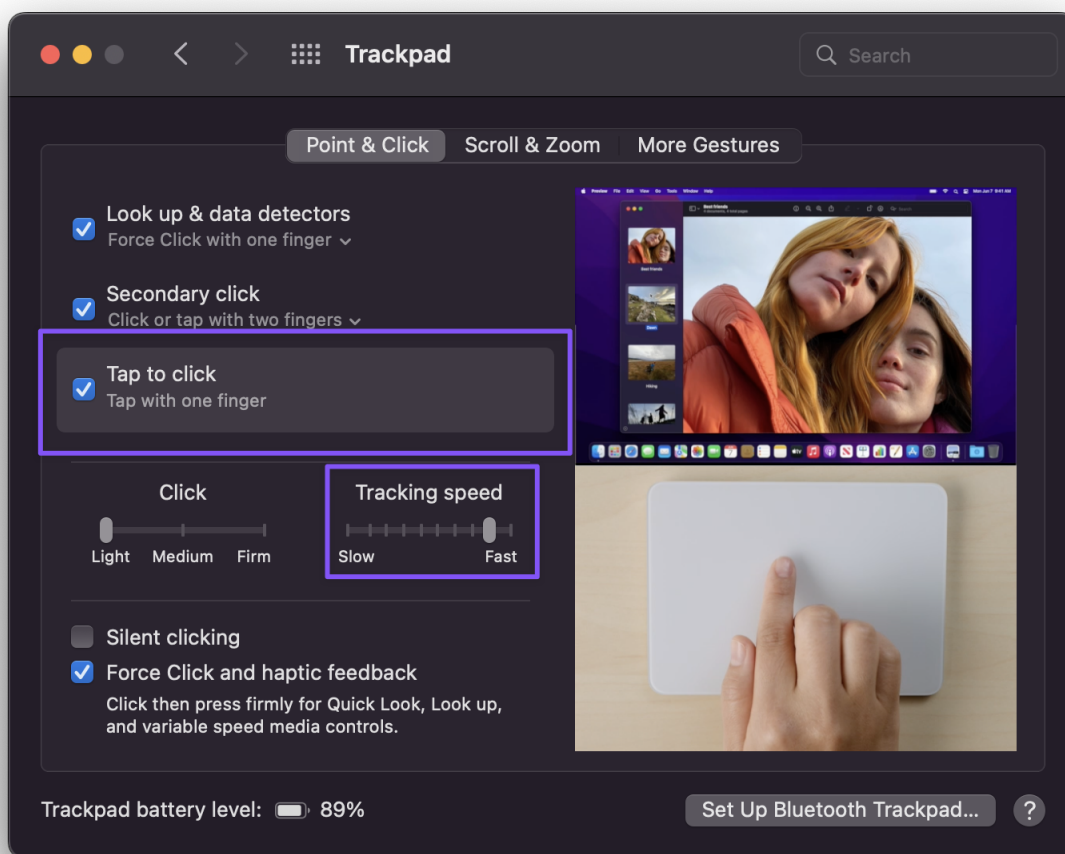
Ostatnim godnym uwagi jest skrót przełączający tryb pełnoekranowy, z którego bardzo często zdarza mi się korzystać, zamiast wciskać przycisk w lewym górnym rogu okna:



Warto ustawić te skróty, ponieważ będziemy z nich jeszcze korzystać.

Ustawienia trackpada

W przypadku trackpada są dwa ustawienia, które definitywnie potrzebujesz zmienić. Jedno z nich to kliknięcie "tapnięciem" (nie musisz wciskać trackpada aby kliknąć, wystarczy lekko stuknąć). A drugie dotyczy przyspieszenia kursora na niemal maksimum. Będziesz potrzebować chwili aby się do tego przyzwyczaić ale warto.



Kilka ustawień systemowych

Wyświetlanie ukrytych plików:

```
defaults write com.apple.finder AppleShowAllFiles YES  
killall Finder
```

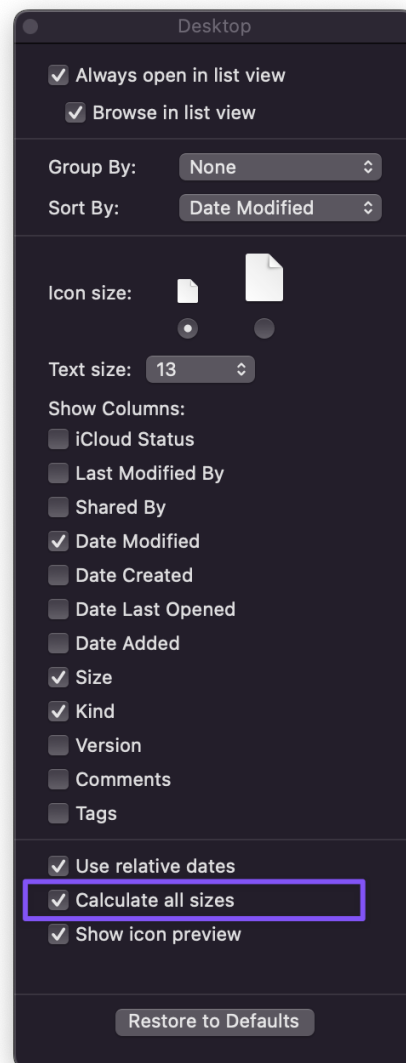
Ukrycie plików wyświetlanych na pulpicie (niebawem pokażę Ci jak łatwo się do nich dostać)

```
defaults write com.apple.finder CreateDesktop -bool false  
kilall Finder
```

Teraz uruchom Finder i wybierz te dwie opcje z menu **View** aktywuj opcje **Show Path Bar** oraz **Show Status Bar**.

Jedna z nich wyświetli ścieżkę do bieżącego katalogu a druga wyświetli status bar wyświetlający kilka przydatnych informacji.

Teraz będąc w Finderze wciśnij **⌘J** aby otworzyć ustawienia widoku i zaznacz opcję **Calculate all sizes**, która będzie odpowiedzialna za przeliczanie rozmiarów katalogów.



W kontekście Findera bardzo przydaje się też skrót **⌘C** (skopiowanie ścieżki bieżącego katalogu) oraz **⇧⌘C** (skopiowanie ścieżki aktualnie zaznaczonego pliku). Zapamiętaj to, ponieważ może przydać Ci się na potrzeby tworzenia późniejszych makr.

Ustawienia git

Jeżeli jeszcze tego nie masz, zmień globalne ustawienia git:

```
git config --global user.name "Your Name"
git config --global user.email "you@your-domain.com"
```

Dodaj alias dla git log (możesz go użyć poleceniem git lg):

```
git config --global alias.lg "log --color --graph --pretty=format:'%Cred%h%Creset -%C(yellow)%d%Creset %s %Cgreen(%cr) %C(bold blue)<an>%Creset' --abbrev-commit"
```

Ustaw domyślny branch z master na main:

```
git config --global init.defaultBranch main
```

Zawsze trzymaj się zasady konfigurowania globalnych ustawień narzędzi z których korzystasz. To samo ma zastosowanie np. do plików package.json. Unikaj sytuacji w których powtarzasz wielokrotnie te same rzeczy, w szczególności gdy są tak oczywiste jak wpisanie swojego imienia i maila.

Ustawienia terminala

Jeżeli jeszcze tego nie masz, zainstaluj Homebrew jako swój Package Manager:

```
/bin/bash -c "$(curl -fsSL <https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh>)"
```

```
brew update
```

I zainstaluj poniższe dodatkowe narzędzia do terminala:

```
brew install \\  
  zsh \\  
  wget \\  
  exa \\  
  git \\  
  nvm \\  
  pnpm \\  
  - alternatywa dla bash  
  - alternatywa dla CURL  
  - alternatywa dla polecenia ls  
  - git (jeżeli jeszcze nie masz)  
  - zarządzanie wersjami Node.js  
  - package manager. Niebawem warto sięgnąć po bun
```

```
graphicsmagick \ - manipulacja zdjęciami  
gh \ - cli dla Github  
vips - bardziej wydajna alternatywa dla graphicsmagick
```

Zainstaluj też Oh My Zsh, czyli framework ułatwiający zarządzanie konfiguracją ZSH:

```
wget <https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/tools/install.sh>  
sh install.sh
```

Po jego instalacji w pliku `~/.zshenv` możesz umieścić aliasy, które będą ułatwiały Ci posługiwanie się terminalem. W moim przypadku są to aliasy powiązane z często wykonywanymi przeze mnie akcjami (np. łączenie się przez ssh z moim serwerem).

Dla przykładu, aplikacja iA Writer w której piszę niemal wszystkie teksty, nie posiada CLI. Dlatego utworzyłem prosty alias, który umożliwia mi otwieranie plików poleceniem

`ia [nazwapliku]`:

A screenshot of a code editor window titled ".zshenv" with a search icon. The file content shows two lines: line 1: "alias ia='open \$1 -a /Applications/iA\ Writer.app'" and line 2: an empty line.

```
< > .zshenv x  
1 |alias ia="open $1 -a /Applications/iA\ Writer.app"  
2
```

To tylko jedna linia, którą wykorzystuję bardzo często zarówno bezpośrednio w terminalu jak i poprzez makra, o których napiszę więcej później.

Ustawienia SSH

Jeżeli jeszcze tego nie masz, utwórz katalog .ssh:

```
mkdir ~/.ssh
```

oraz utwórz parę kluczy SSH (publiczny i prywatny):

```
ssh-keygen -t ed25519 -C "twojemail"
```

Klucz te przydadzą nam się później do nawiązywania połączenia z naszym serwerem VPS. Możesz też wykorzystać klucz publiczny na swoim koncie Github. Warto dodać,

że całkiem dobrym pomysłem jest generowanie różnych par kluczy dla różnych usług. To na co się zdecydujesz zależy od Ciebie.

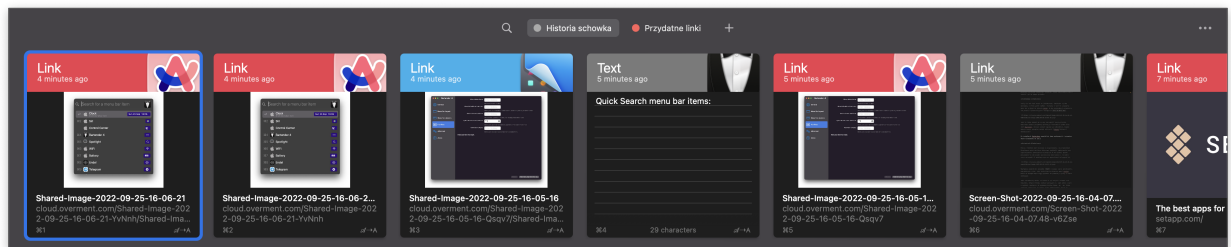
Ostatnim krokiem będzie dodanie Twojego klucza do `keychain` systemu MacOS

```
ssh-add --apple-use-keychain ~/.ssh/id_rsa
```

Pamiętaj że wykorzystanie kluczy SSH jest zdecydowanie bardziej bezpieczne niż posługiwanie się hasłem. Musisz tylko dbać o to aby Twój klucz prywatny pozostał w Twoich rękach.

Manager Schowka

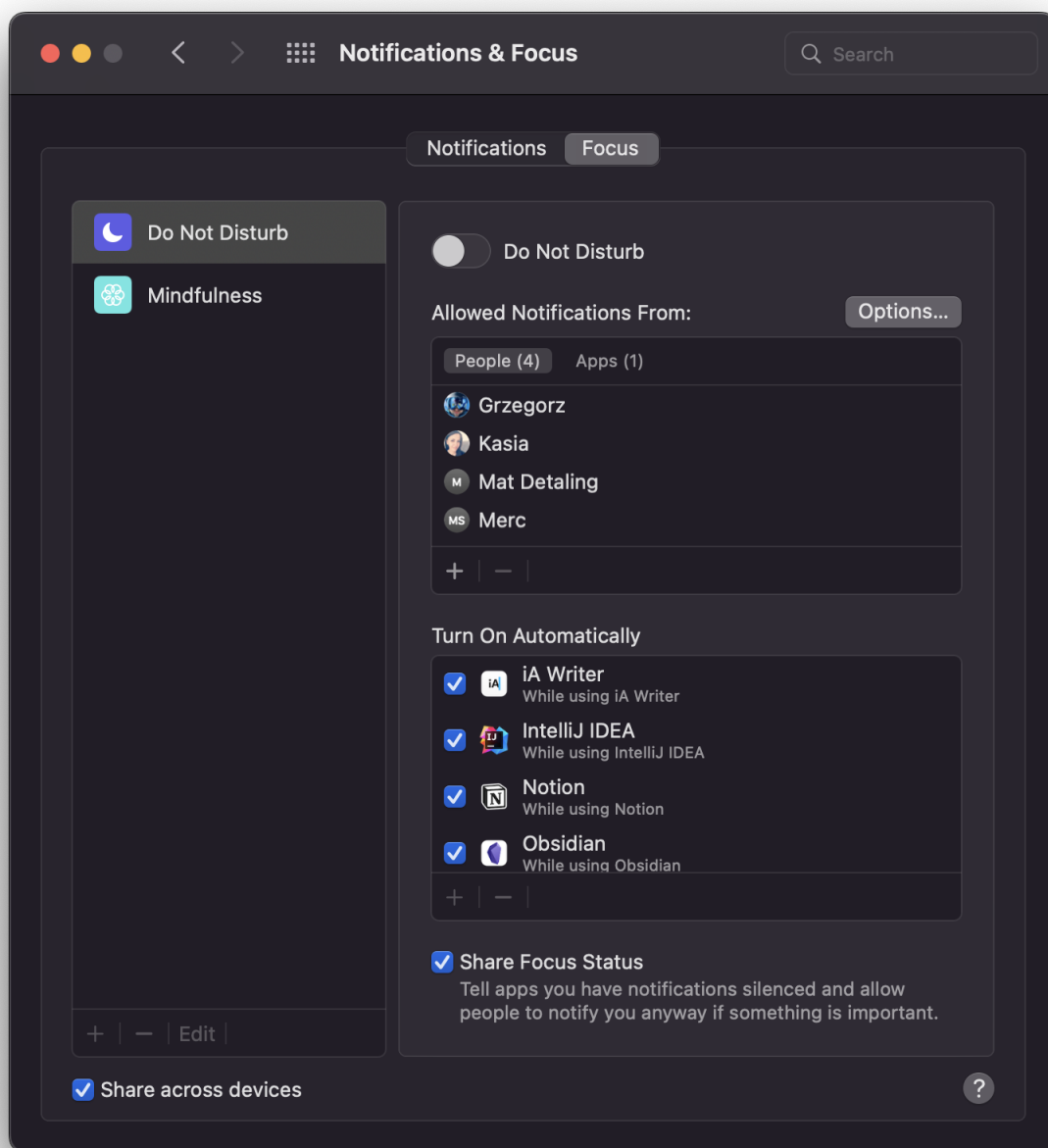
Kolejną aplikacją z pakietu Setapp, która jest bardzo warta Twojej uwagi to Paste. Jest to nic innego jak manager schowka, który wygląda tak:



Wystarczy że po jego instalacji wciśniesz `⌘⇧V` aby aktywować okno podobne do tego powyżej. Znajdziesz tam wszystkie skopiowane przez siebie wartości z wyjątkiem haseł pochodzących np. z 1Password. Zwróć tutaj uwagę na możliwość zapisywania wybranych wartości, poprzez przypisanie ich do konkretnej kategorii. Bardzo polecam Ci zapisać w ten sposób rzeczy, które zdarza Ci się kopiować najczęściej (np. kolory, logotypy, zdjęcia).

Ustawienia trybu skupienia

Tryb skupienia jest szczególnie ważny podczas programowania. W ustawieniach `Notifications & Focus` znajdziesz zakładkę `Focus` w której możesz przypisać aplikacje które będą aktywować ten tryb automatycznie oraz aplikacje i kontakty które nawet w tym trybie będą mogły się z Tobą komunikować:



Jeżeli zaznaczysz dostępną na dole opcję `Share across devices` tryb ten będzie dla Ciebie dostępny na dowolnym urządzeniu Apple jakie posiadasz.

CLI i Skrypty Shell

Pierwszym elementem, który z jednej strony daje niesamowite możliwości a jednocześnie opiera się o wykorzystywanie Twoich programistycznych umiejętności jest

CLI (Command Line Interface) oraz pisanie skryptów Shell (uruchamianych np. w terminalu).

Nie wiem czy zdajesz sobie z tego sprawę, ale możesz wykorzystać nawet node.js do pisania takich skryptów. W dodatku niektóre programy oraz usług same w sobie oferują wersję CLI.

Przykładem jednej z takich usług jest Remove.bg z pomocą której usuniesz tło ze zdjęcia: <https://github.com/remove-bg/remove-bg-cli>. Podobnie też mamy Dropshare (dostępna w pakiecie Setapp).

Alternatywnie nic nie stoi na przeszkodzie aby wykorzystać wget lub curl do tego aby podłączyć się do usługi udostępniającej API w celu przygotowania własnego CLI.

Jeżeli zastanawiasz się dlaczego w ogóle potrzebne Ci CLI skoro wiele aplikacji udostępnia GUI? Odpowiedź to: efektywność. Przykładowo, czy wolisz przechodzić do strony TinyPNG aby zoptymalizować obrazek, czy wcisnąć skrót klawiszowy na zaznaczonym pliku aby niemal natychmiast otrzymać jego zoptymalizowaną wersję? Albo wolisz przechodzić do tłumacza Deepl aby przetłumaczyć wybraną frazę, czy z pomocą jednego skrótu klawiszowego natychmiast przetłumaczyć zaznaczony tekst?

Poza tym wykorzystując fakt, że prawdopodobnie znasz JavaScript oraz przynajmniej podstawy Node.js, możesz pisać własne skrypty oraz tworzyć CLI. Istnieje szereg narzędzi, które ułatwią Ci projektowanie takich skryptów. Mam tutaj na myśli minimist / prompt / ora / boxen / chalk oraz przede wszystkim zx, z pomocą którego wykonasz skrypty Shell bezpośrednio w kodzie JavaScript i to ze wsparciem składni async / await!

Pomysły dotyczące wykorzystania CLI:

- charakterystyczne dla Ciebie modyfikacje plików (np. Obsidian)
- transformacja obrazów i filmów (np. optymalizacja czy usuwanie tła)
- transformacja tekstu (np. Markdown to Styled Text lub HTML)
- łatwiejsze wykonywanie złożonych poleceń shell
- wykonywanie poleceń shell uwzględniających dane od użytkownika
- połączenie z zewnętrznymi usługami udostępniającymi API (np. wspomniane TinyPNG czy Deepl)
- połączenie z własnymi mikro-serwisami

- wykorzystanie CLI aplikacji desktopowych (np. IntelliJ)

Jak widzisz możliwości są tutaj bardzo duże, jednak faktycznie potrzebujesz trochę czasu i kreatywności aby przygotować skrypty, które będą Ci służyć i oszczędzać Twój czas. Pamiętaj też że w wielu przypadkach dobrym pomysłem będzie wykorzystanie skryptów przygotowanych przez innych (szukaj na Github).

No i ostatecznie bardzo dobrym pomysłem jest wykorzystanie środowiska, które już znasz. Domyślam się że w większości przypadków będzie to ekosystem JavaScriptu i Node.js, jednak chociażby Python również sprawdzi się tutaj bardzo dobrze. Także jeżeli tylko możesz, **wykorzystuj przewagi które już masz, aby z ich pomocą budować kolejne.**

Przykładowo jeżeli nie kojarzysz składni CURL ale znasz składnię Fetch, możesz skorzystać z converterów, które ułatwią Ci ich zamianę (np. [pierwszy tool z brzegu](#)). Ponownie - ułatwaj sobie pracę tam gdzie to możliwe. Pamiętaj że tutaj priorytetem jest to, aby opracować rozwiązanie które będzie Ci służyć i oszczędzało czas oraz energię.

Okazuje się jednak, że to wszystko można przenieść na kolejny poziom, ponieważ skrypty shell mogą komunikować się nie tylko z systemem plików czy zewnętrznymi usługami, ale także mogą bezpośrednio komunikować się z MacOS z pomocą...AppleScript!

Gdyby tego było mało, to mam do tego jeszcze jeden dodatek ale do tego jeszcze dojdziemy.

Zaprogramuj swojego Maca!

AppleScript to skryptowy język umożliwiający wchodzenie w interakcję z interfejsem systemu oraz aplikacji desktopowych. Z jego pomocą możesz symulować wciśnięcia przycisków, uruchamiać aplikacje, wchodzić w interakcję ze schowkiem, pobierać informacje o pozycjach okien a nawet je modyfikować.

Bardzo ważną informacją jest fakt, że z pomocą paczki `[run-applescript]`

([<https://github.com/sindresorhus/run-applescript>](https://github.com/sindresorhus/run-applescript)) możesz wykonywać skrypty

AppleScript bezpośrednio z poziomu skryptów node.js, które mogą wykonywać skrypty Shell!

W rezultacie otrzymujemy tutaj połączenie pomiędzy ekosystemem JavaScriptu, możliwościami Shella oraz interakcjami z Systemem macOS! W tym momencie jedyne czego potrzebujemy to kreatywność i trochę wolnego czasu.

Wyobraź sobie przykładowe scenariusze:

- uruchamiasz skrypt
- wszystkie aplikacje zostają zamknięte
- otwiera się IDE i terminal
- uruchamia polecenia startujące Twój projekt
- otwiera w przeglądarce strony, których potrzebujesz
- uruchamia się Twoja ulubiona muzyka
- zaczynasz pracę

albo drugi scenariusz:

- zaznaczasz plik
- który zostaje przesłany natychmiast do chmury
- jego adres ląduje w Twoim schowku
- otwiera się Twój klient poczty
- oraz okno tworzenia nowej wiadomości
- w której treści pojawia się skopiowany link

albo trzeci scenariusz:

- zaznaczasz fragment ekranu (nie tekst, fragment.)
- system OCR odczytuje tekst z zaznaczonego fragmentu
- przesyła go do DeepL
- i otrzymujesz jego tłumaczenie
- tłumaczenie trafia jako karta w aplikacji Anki

Mało? Teraz Twoja kolej.

Jeżeli potrzebujesz inspiracji, sprawdź sobie repozytoria pod hasłem CLI, API usług z których korzystasz oraz przykłady skryptów AppleScript tworzonych przez społeczność.

Zasada jest tutaj bardzo prosta: patrzysz na to, co robisz, z jakich narzędzi korzystasz i w jaki sposób oraz zadajesz sobie pytanie "czy warto stworzyć skrypt, który Ci to

ułatwi"? A potem zastanawiasz się jak możesz go przygotować, sięgając po wszystkie dostępne możliwości i ich kreatywne połączenie.

Przykładem takiego kreatywnego połączenia może być wykorzystanie skryptu Node.js w którym sięgniesz po AppleScript, który przekaże do Twojej przeglądarki kod JavaScript i wykona go! W ten sposób możesz np. pobierać dane z aktualnie otwartych stron lub nawet wykonywać akcje w postaci kliknięć wybranych elementów.

Przykładowo w moim przypadku wystarczy taki snippet:

```
tell application "Arc"
    set JSResult to execute front window's active tab javascript JSSnippet
end tell
```

Aby wykonać dowolny skrypt przypisany do zmiennej JSSnippet. W ten sposób ułatwiam sobie wykonywanie różnego rodzaju akcji w używanych przeze mnie serwisach i aplikacjach. Podobnie też wykorzystuję drugi skrypt do tego aby odczytać bieżący tytuł oraz URL otwartej strony i przesłać go do Notion:

```
tell application "Arc"
    set currentURL to URL of active tab of window 1
    set currentTitle to title of active tab of window 1
end tell
```

Efekt końcowy, połączony z moimi dodatkowymi skryptami wygląda tak, że w chwili gdy znajduję się w przeglądarce i wykonam tapnięcie dwoma palcami, pojawia się takie oto okno:



What book would you like to add?

Add a title and optional description

Title

Home | Ahoy!

URL

<https://community.ahoy.so/home>

Nevermind

Save

Formularz który widzisz został uzupełniony automatycznie z pomocą powyższego AppleScript a po jego wysłaniu, link i tytuł zostaną zapisane w Notion oraz aplikacji Matter w której czytam zapisywane w ten sposób artykuły.

Jak widzisz możliwości o których tutaj mówimy są duże i początkowo mogą przytłaczać. Nie ma w tym nic dziwnego ale jednocześnie jest na to proste rozwiązanie: **wdrażanie zmian bardzo małymi krokami**. Czas płynie bardzo szybko. Wystarczy zadbać o to

aby regularnie dodawać nawet coś małego do swoich aktywności. Czasem to będzie nowy skrót, innym razem aplikacja a jeszcze innym razem cała automatyzacja lub mikro-serwis.

Niestety na tym etapie nie mam do polecenia żadnych sensownych materiałów na temat AppleScript. Sam opieram się głównie o przykłady z Internetu, które modyfikuję do swoich potrzeb. Na szczęście niemal zawsze jest to w pełni wystarczające.

Nie wiem czy jest to już na tym etapie dla Ciebie jasne, ale widać już w jaki sposób zaczynają łączyć się ze sobą te wszystkie elementy o których mówię.

Wykorzystaj sprytne narzędzia

Będę z Tobą szczery. Pisanie wszystkiego od absolutnych podstaw nie tylko jest dość wymagające ale przede wszystkim zajmuje czas. Wszystko o czym tutaj piszę ukierunkowane jest na oszczędność Twojej energii oraz właśnie czasu. Z tego powodu sytuacja w której wykorzystasz bardzo dużo swoich zasobów na opracowanie pozornie prostego mechanizmu, trochę mija się z celem. Dlatego należy szukać dróg na skróty, poprzez wykorzystanie sprytnych narzędzi.

Mam tutaj na myśli przede wszystkim aplikacje Raycast oraz Keyboard Maestro.

Raycast to nic innego jak Launcher, który całkowicie zastąpi Ci systemową wyszukiwarkę Spotlight i doda mnóstwo nowych elementów. W połączeniu z Twoją programistyczną wiedzą, przeniesie Twój workflow na kolejny poziom.

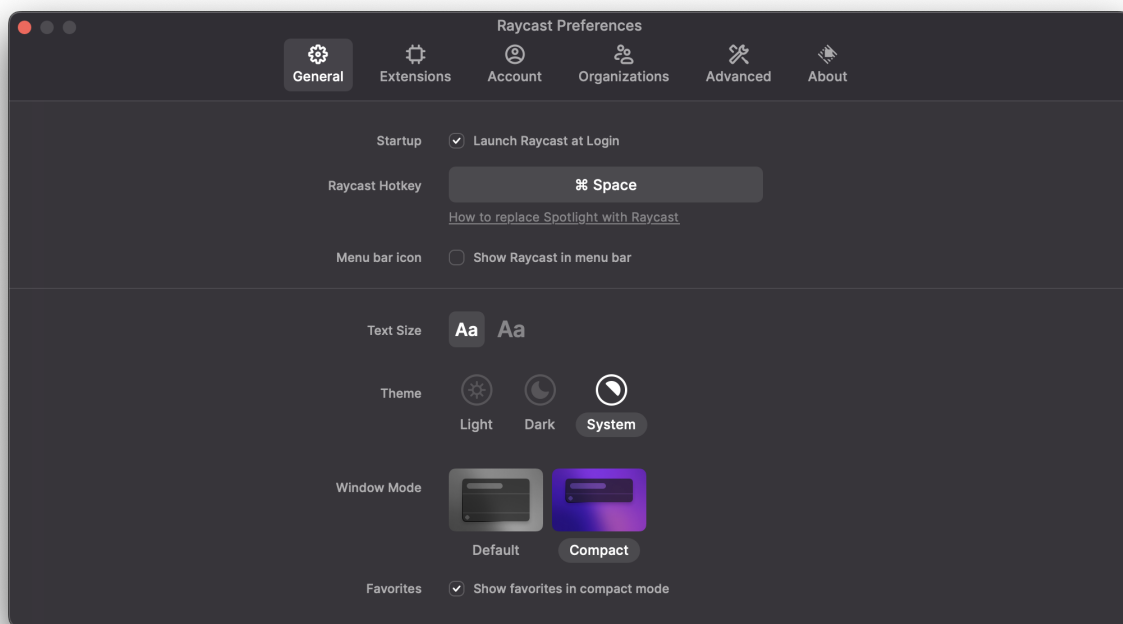
Niewykluczone że znasz aplikację Alfred, z której sam korzystałem przez wiele lat. Potraktuj Raycast jak nową generację tego rodzaju narzędzi.

Gdyby tego było mało, w MacOS masz do dyspozycji genialną aplikację jaką jest Keyboard Maestro. Ta niepozorna aplikacja wyglądem przypominająca lata 90-te miała swoją ostatnią aktualizację na przestrzeni ostatnich dwóch tygodni i oferuje więcej niż jestem w stanie sobie wyobrazić.

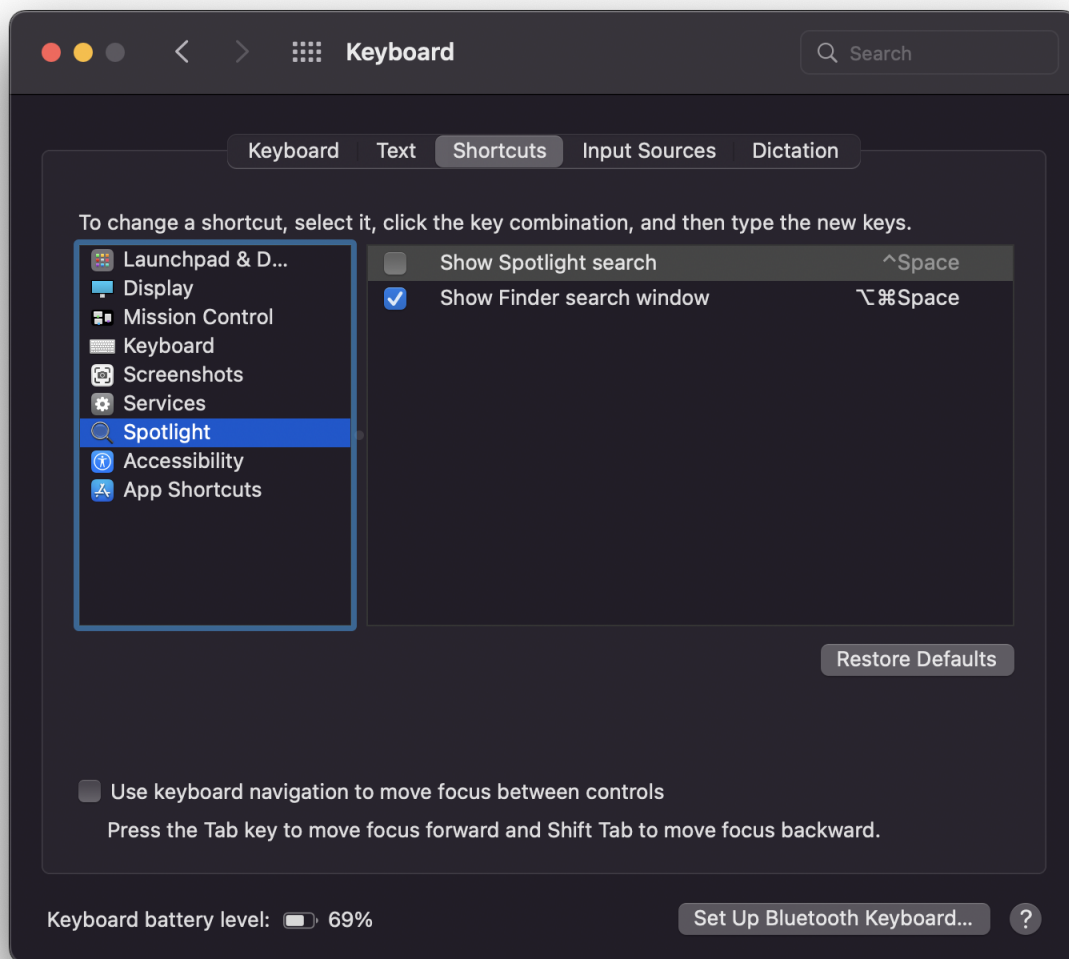
Pokażę Ci teraz jak je w pełni wykorzystywać na swoją korzyść.

Launcher

Przejdź na raycast.com, pobierz i zainstaluj aplikację. Po jej uruchomieniu przejdź do preferencji (⌘,) i upewnij się że skrót klawiszowy odpowiedzialny za jej włączenie to  **Spacja**



Następnie przejdź do Preferencji Systemowych i w ustawieniach klawiatury w zakładce **Shortcuts** wyłącz skrót odpowiedzialny za uruchamianie Spotlight:



Teraz wciśnij **⌘ Spacja** aby włączyć Raycast. Wpisz Store i przejdź przez listę dostępnych rozszerzeń. Zainstaluj te, które są powiązane z aplikacjami z których korzystasz. Tutaj masz całą moją listę:

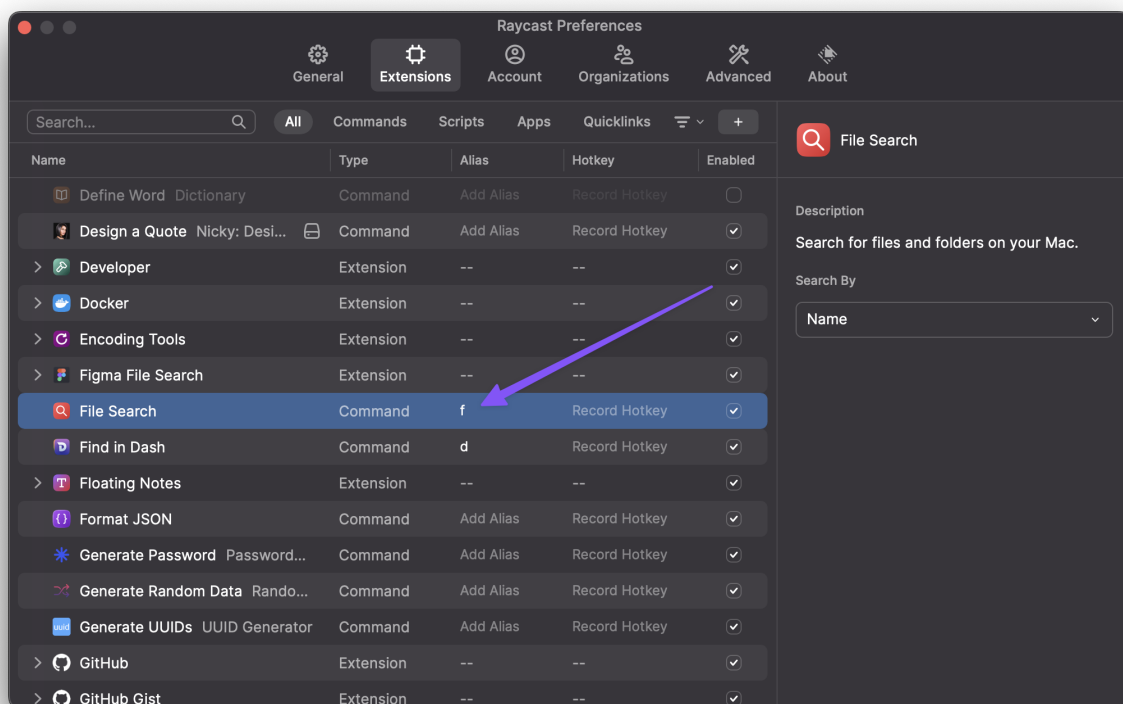
- Base64
- Check Diff
- Check Today's Hunts
- Color Search (Tailwind)
- Conventional Comments
- Create Meet

- cURL
- Developer
- Docker
- Encoding Tools
- File Search
- Find in Dash
- Floating Notes
- Format JSON
- Generate Password
- Generate Random Data
- Generate UUIDs
- Github
- Github Gist
- Google Workspace
- Hacker News
- Kill Port
- Kill Process
- Life Progress
- List Macros (Keyboard Maestro)
- List Repos
- Lorem Ipsum
- Manage Files
- Navigation
- Netlify
- Notion
- Obsidian

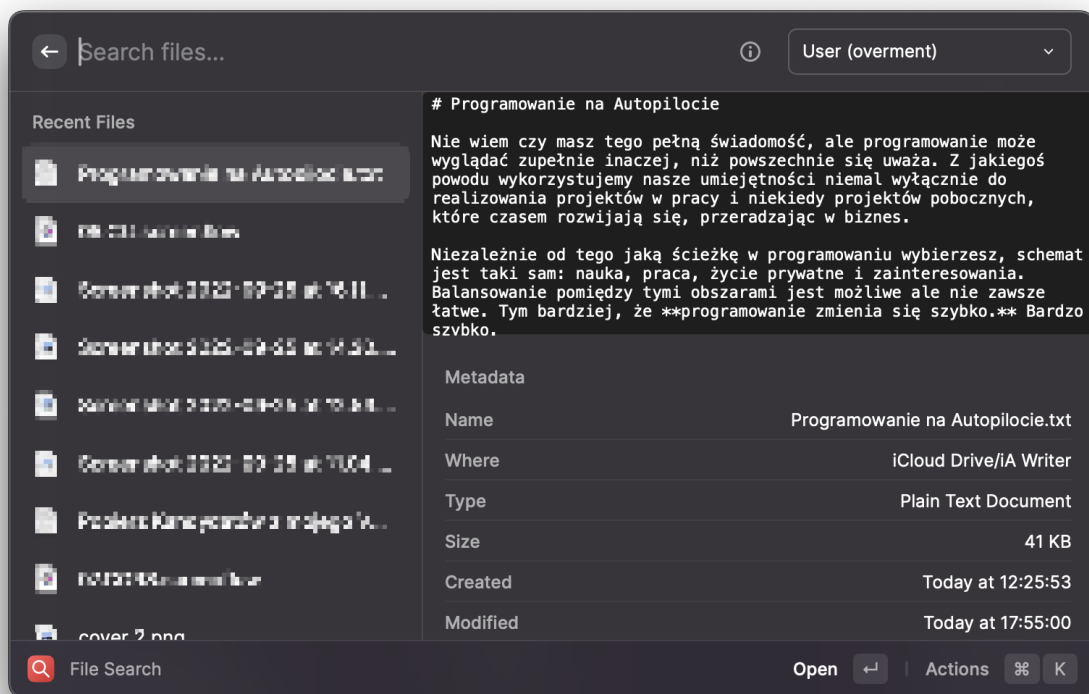
- Pipe Commands
- Quick Search
- Quicklinks
- Ray.so
- Remove Background From Image
- Search Menu Items
- Search Reddit
- Search Repositories
- Search Web Docs (MDN)
- Shortcuts
- Slack
- Spotify Controls
- Tailwind CSS
- Todoist
- Window Management
- Zoom

Myślę że wiele z powyższych rozszerzeń możesz z powodzeniem wykorzystać u siebie. Tymczasem na Twoją szczególną uwagę zasługuje możliwość przypisania skrótów klawiszowych oraz aliasów. W ten sposób uruchomisz je natychmiast:

Przykładowo tutaj mam przypisany alias do wyszukiwania plików.



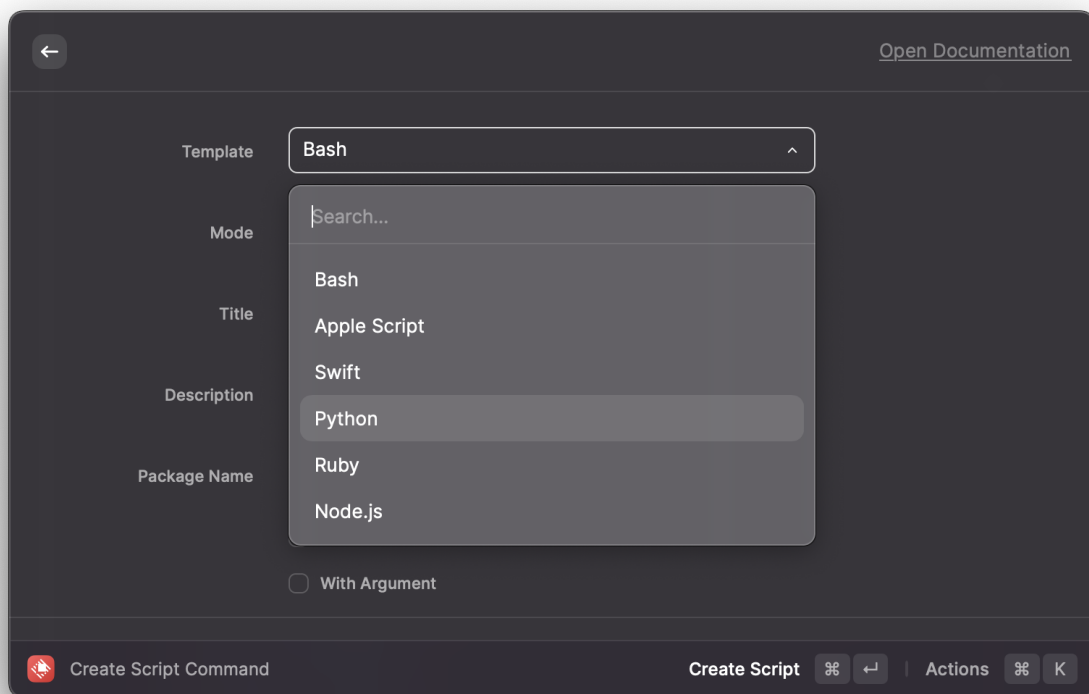
Jeżeli teraz włączę Raycast i wcisnę  oraz spację, zobaczę listę moich plików z opcją wyszukiwania:



Własne polecenia

Prawdziwe możliwości Raycasta zaczynają się ujawniać w momencie gdy skorzystasz z opcji: `Create Script Command` z pomocą której, jak sama nazwa wskazuje, utworzysz komendę która uruchomi dla Ciebie skrypt.

Taki skrypt możesz przygotować praktycznie w dowolnej technologii. Przypomnij sobie w tym momencie `run-applescript` oraz `zx` dostępne w node.js i myślę że wybór będzie dla Ciebie prosty.



Tutaj możesz więc tworzyć własne skrypty a następnie uruchamiać je albo z pomocą Raycast albo z pomocą skrótu klawiszowego.

Snippets

Drugim istotnym z programistycznego punktu widzenia rozszerzeniem Raycast jest **Manage Snippets** z pomocą którego możesz zapisywać często wykorzystywane snippets.

← Learn More

Organization ☐ None

Name

Snippet

Find your inner Shakespeare...

B I U | 🔗 🔗 +

Keyword ⓘ

Create Snippet Create Snippet ⌨️ ↩

Wyrobiecie nawyku ich tworzenia oraz wykorzystywania sprawdzi się świetnie zarówno w Twojej codziennej pracy, jak i na potrzeby tworzenia rozszerzeń i własnych automatyzacji. Polecam Ci przynajmniej spróbować a najlepiej od razu dodać kilka przydatnych skryptów, które wpadają Ci w tym momencie do głowy.

API Raycast

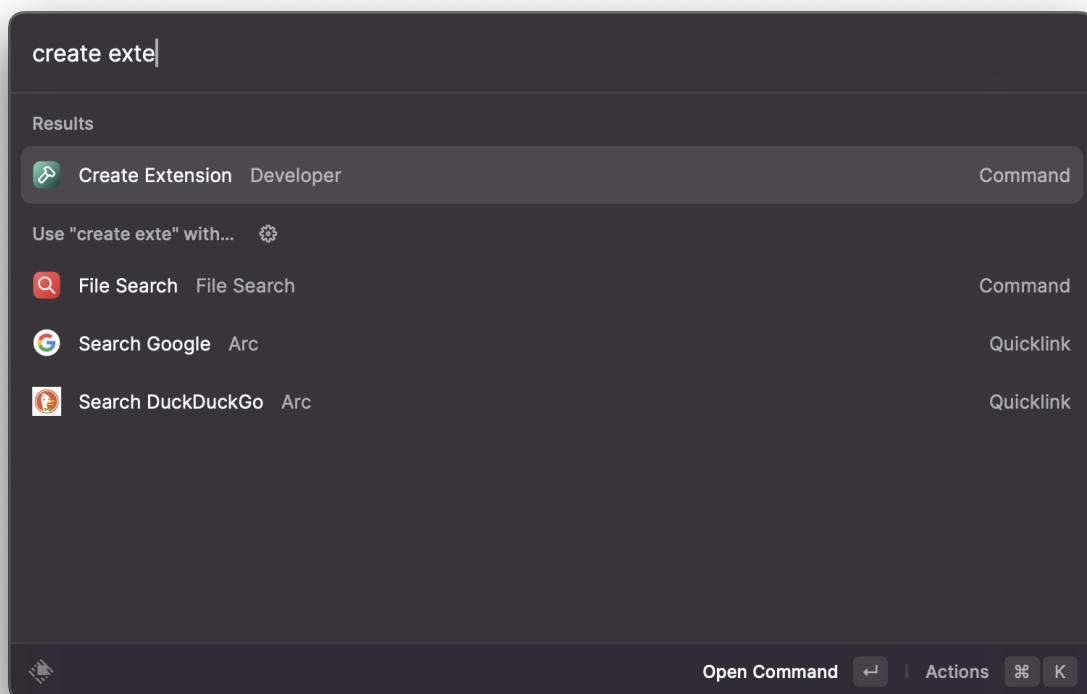
No i na koniec mam dla Ciebie absolutny hit, jakim jest tworzenie własnych rozszerzeń Raycast z pomocą Node.js, React i TypeScript!!!

Na początek przejrzyj sobie dokumentację, która moim zdaniem jest genialnie przygotowana i dobrze sprawdza się w praktyce:

<https://developers.raycast.com>



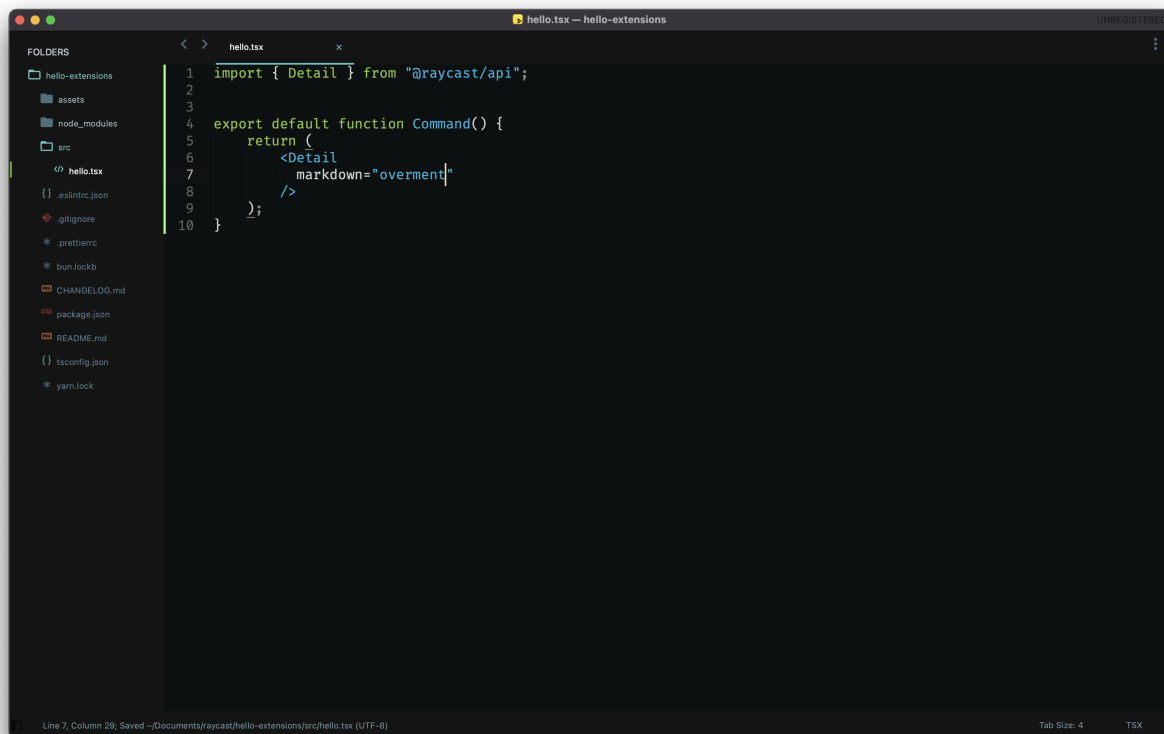
Utworzenie Twojego pierwszego rozszerzenia jest niezwykle proste. Przede wszystkim musisz utworzyć nowe rozszerzenie korzystając z akcji [Create Extension](#)



Następnie wypełnij formularz podając nazwę rozszerzenia jakie chcesz utworzyć oraz oraz lokalizację w której zostaną zapisane jego pliki. Utwórz też pierwsze polecenie (Command) i wciśnij `⌘ Enter`

Pliki rozszerzenia zostaną utworzone. Otwórz je w swoim edytorze i w terminalu uruchom `yarn install && yarn dev`

W pliku `src/index.tsx` zapisz taki kod:



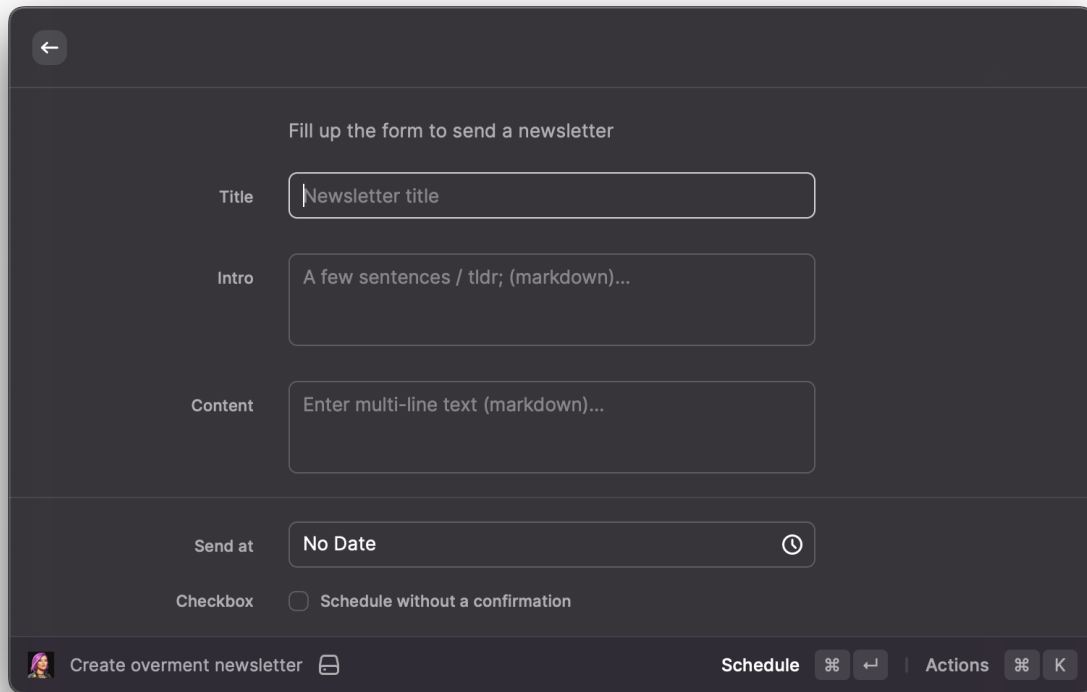
```
1 import { Detail } from "@raycast/api";
2
3
4 export default function Command() {
5   return (
6     <Detail
7       markdown="overment"
8     />
9   );
10 }
```

A następnie uruchom Raycast. Na górze listy powinno być widoczne Twoje polecenie. Po jego uruchomieniu zobaczysz tekst "overment".

Na tym etapie polecam Ci przede wszystkim:

- dołączyć do Slacka Raycast, gdzie funkcjonuje bardzo aktywna społeczność osób tworzących rozszerzenia
- przejrzeć przykłady z dokumentacji oraz repozytorium raycast
- obejrzeć kilka filmów na YouTube pokazujących jak tworzyć własne rozszerzenia Raycast

Mając taką wiedzę, będziesz w stanie tworzyć własne i niesamowicie rozszerzać możliwości swojego systemu. Sam posiadam kilkanaście rozszerzeń. Z pomocą jednego z nich...wysyłam swój newsletter!



The screenshot shows a dark-themed user interface for creating a newsletter. At the top left is a back arrow icon. Below it, the instruction "Fill up the form to send a newsletter" is centered. The form consists of three main text input areas: "Title" (containing "Newsletter title"), "Intro" (containing "A few sentences / tldr; (markdown)..."), and "Content" (containing "Enter multi-line text (markdown)..."). Below these is a "Send at" section with a dropdown menu set to "No Date" and a clock icon. Underneath is a "Checkbox" section with an unchecked box and the text "Schedule without a confirmation". At the bottom, there is a bar with a profile icon, the text "Create overment newsletter", a document icon, a "Schedule" button with a keyboard icon, and an "Actions" button with a keyboard icon and the letter "K".

Od siebie dodam, że całkiem niedawno pojawiła się też możliwość tworzenia rozszerzeń, które mogą być dostępne z poziomu Menu Bar, jednak nie miałem jeszcze okazji żadnego samodzielnie stworzyć.

Keyboard Maestro

Nie jestem w stanie wskazać aplikacji która bardziej zmieniła mój sposób pracy, jak Keyboard Maestro. Aktualnie posiadam niemal 400 makr, które oszczędzają mi wiele godzin pracy każdego miesiąca. Mógłbym napisać kilka e-booków tylko na temat tej aplikacji. Jednak tym razem postaram się przekazać Ci wyłącznie najważniejsze elementy, na które musisz zwrócić uwagę aby wykorzystać potencjał Keyboard Maestro.

Przede wszystkim jest to płatna aplikacja, ale mam kod rabatowy **DESIGNMAESTRO**, który obniży dla Ciebie cenę o 20%.

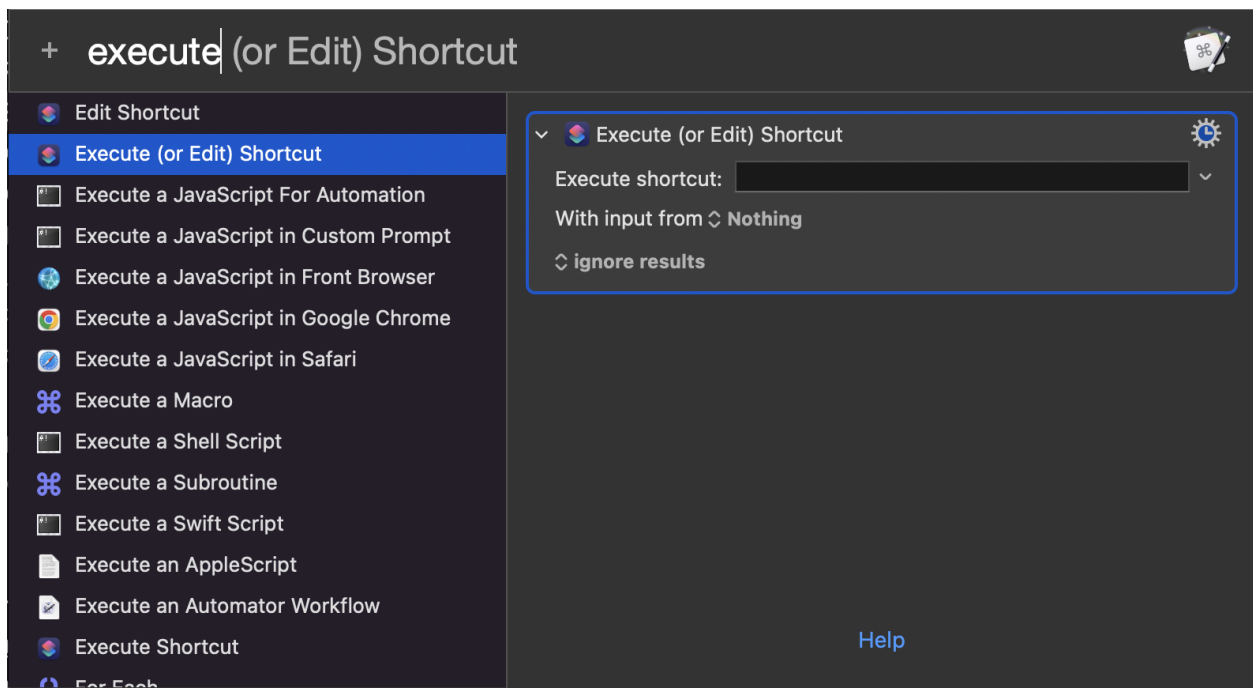
Po instalacji Keyboard Mastro, przyznaj jej niezbędne uprawnienia oraz spróbuj stworzyć swoje pierwsze makro. Zasada jest tutaj bardzo prosta:

- tworzysz grupę makr
- następnie makro
- przypisujesz trigger który uruchomi makro
- oraz definiujesz zestaw akcji, które mają zostać zrealizowane

W momencie gdy wyzwalacz (trigger) zostanie uruchomiony, zostanie wykonany zdefiniowany przez Ciebie zestaw akcji.

Sam schemat jest bardzo prosty i myślę, że bez problemu się w nim odnajdziesz. Jak zwykle największym problemem jest tutaj kreatywność, tworzenie makr oraz ich późniejsze wykorzystanie. Dlatego chciałbym podkreślić kilka szczególnych elementów, które zasługują na Twoją uwagę.

Pierwszy z nich to lista akcji umożliwiających wykonywanie skryptów:



Zwróć uwagę na to, że możesz tutaj wykonywać skrypty JXA (JavaScript for Automation), Shell, AppleScript oraz Swift. Możesz też wykonać kod JavaScript w aktualnie otwartej przeglądarce (w moim przypadku korzystam z przeglądarki Arc i tam konieczne było napisanie własnego AppleScript).

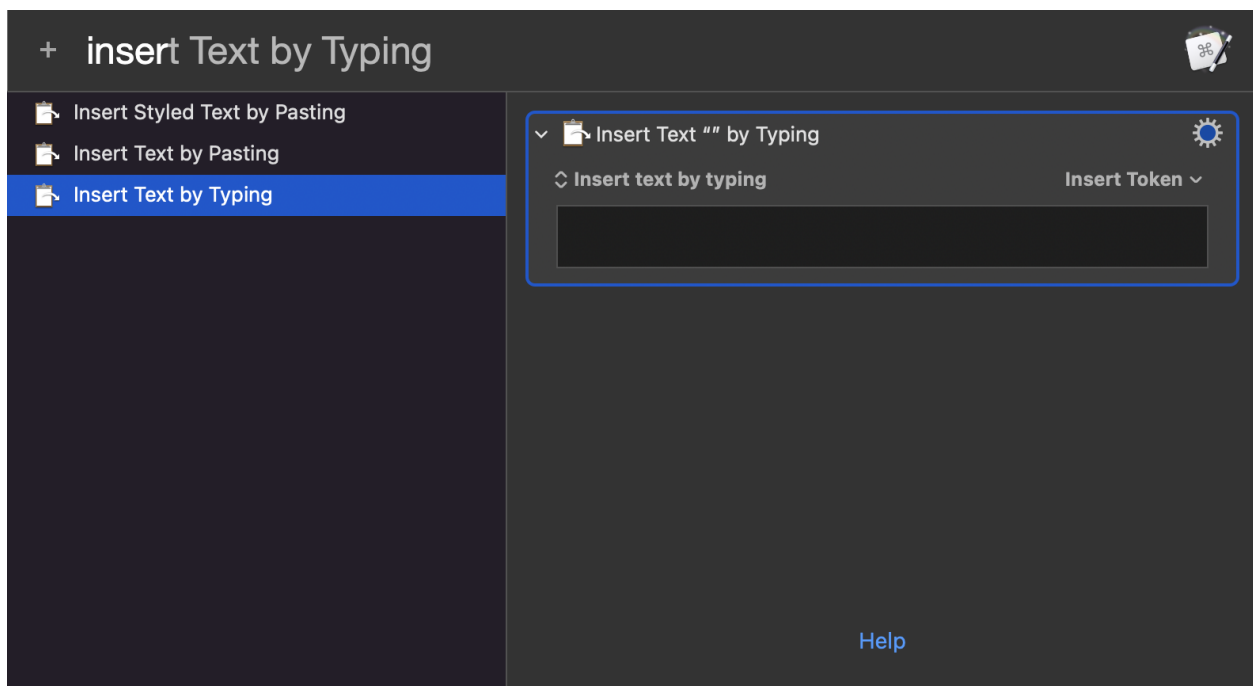
Zatem pierwsza wskazówka mówi Ci o możliwości wykonywania skryptów, które możesz dodać jako akcje do makr.

Wiesz co to oznacza?

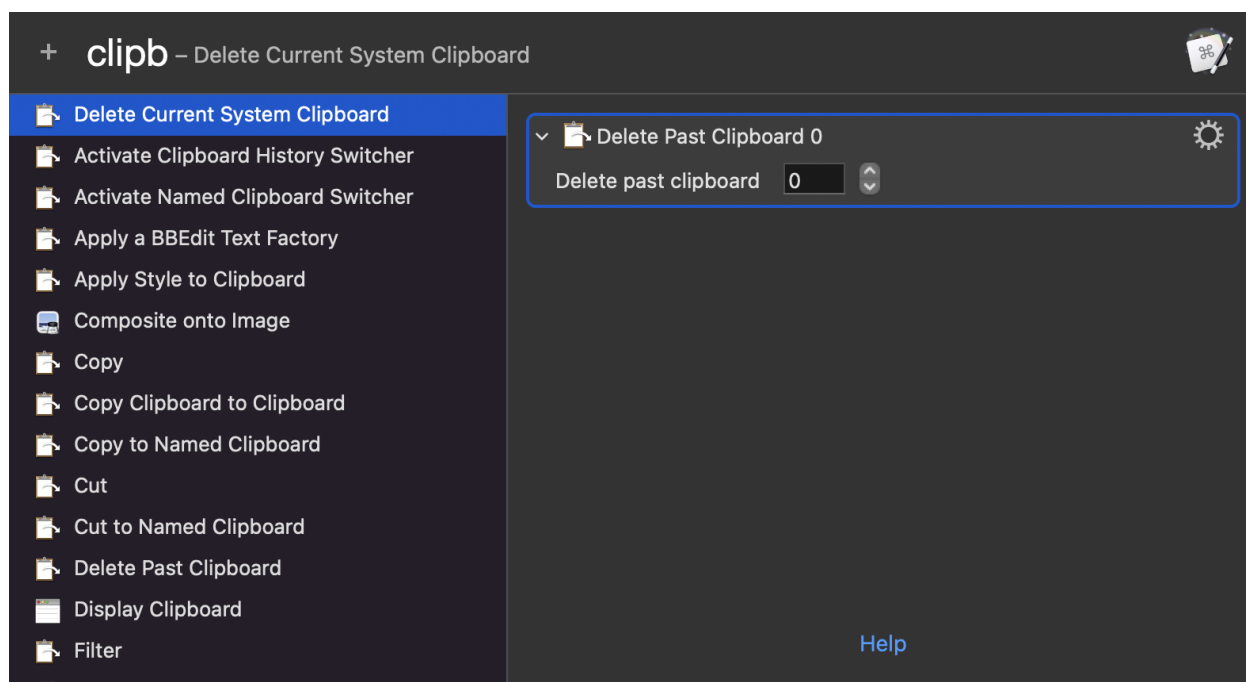
Że możesz łączyć dowolne akcje dostępne w Keyboard Maestro ze swoimi skryptami (i np. przysyłać pomiędzy nimi informacje) oraz przypisywać je do wyzwalaczy (skrótów klawiszowych, określonej godziny czy zdarzenia w Twoim systemie)!

Jest to absolutny hit i coś, co całkowicie wywraca zasady gry. W tej chwili powinien być dla Ciebie jasny pełen potencjał, jaki masz do dyspozycji, w związku z tym, że programujesz.

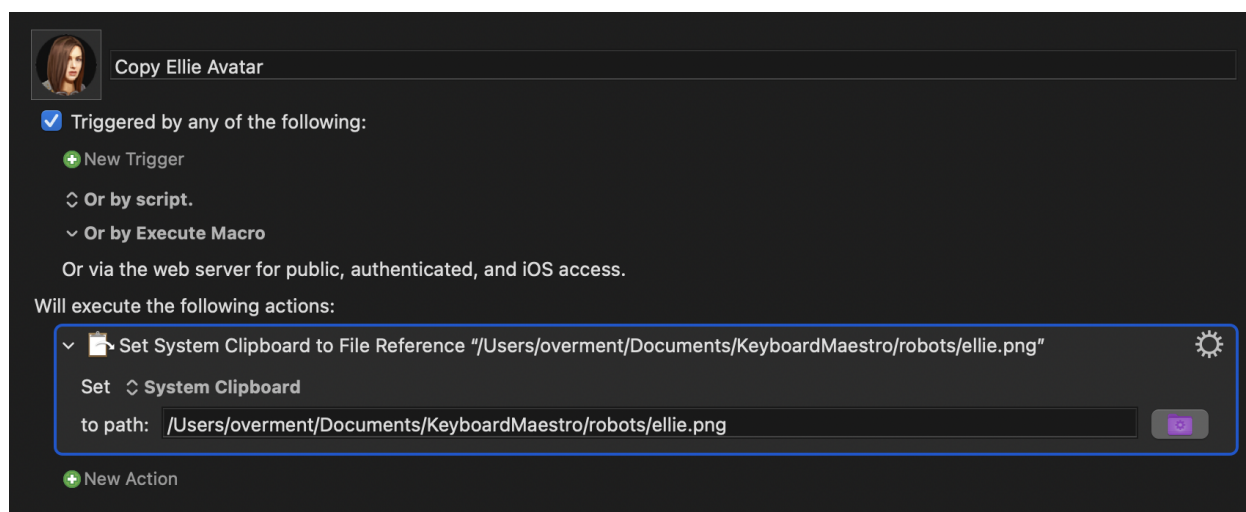
Kolejne akcje, które bardzo zasługują na Twoją uwagę dotyczą wklejania tekstu:



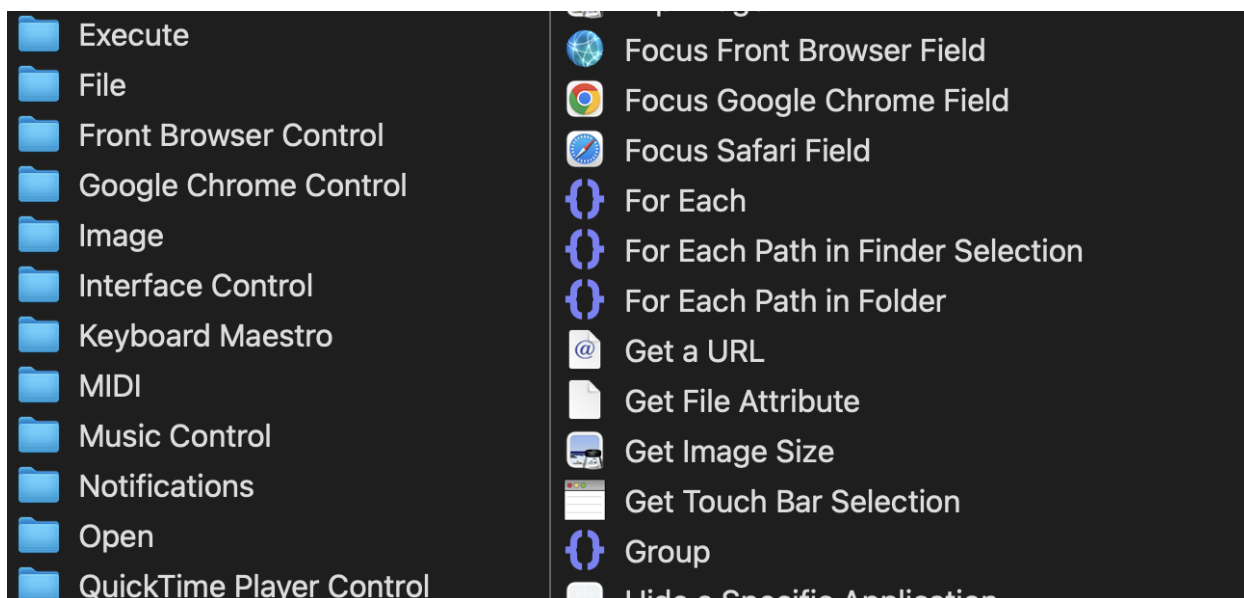
Podobnie też masz możliwość kontroli swojego schowka:



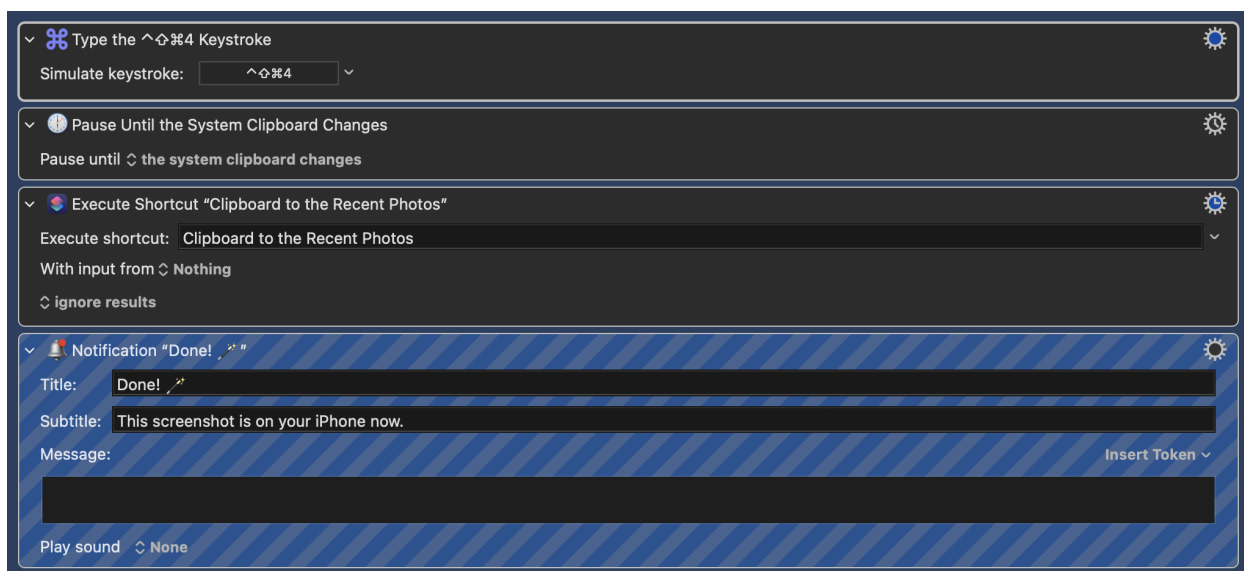
Tutaj możesz nawet przypisać zawartość wybranego pliku do schowka. W ten sposób możesz stworzyć makra, które natychmiast skopiują dane, po które często sięgasz. Spójrz:



A nawet ustawiania zmiennych oraz wykonywania pętli czy instrukcji warunkowych.

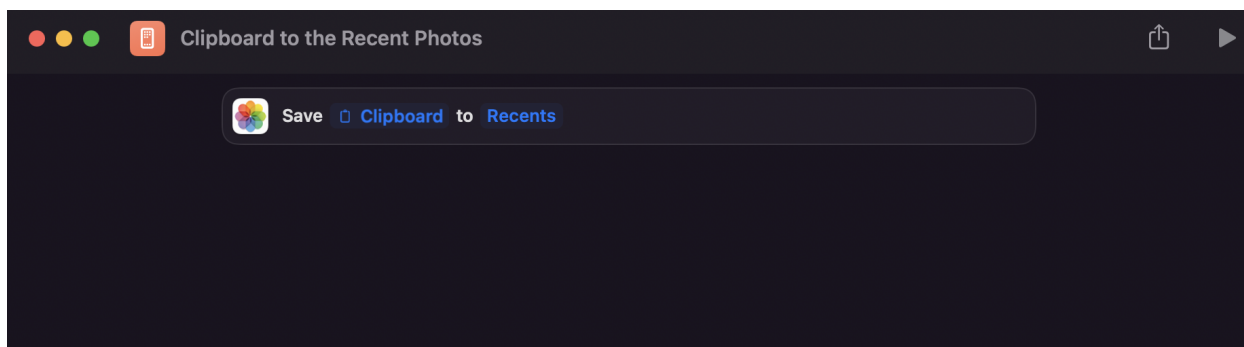


Ciekawostka: Kilka miesięcy temu w Keyboard Maestro pojawiła się opcja uruchamiania automatyzacji Shortcuts. Stworzyłem więc następujące makro:



Jego zadanie polega na tym, aby wywołać skrót klawiszowy odpowiedzialny za "zrobienie zrzutu ekranu i skopiowanie zawartości do schowka". Makro zostaje zatrzymane do czasu gdy zmieni się zawartość schowka lub gdy minie określona przeze mnie liczba sekund. Inaczej mówiąc, gdy zrobię zrzut ekranu, makro wykonuje kolejne akcje w postaci uruchomienia skrótu...który przesyła zawartość schowka do

ostatnich zdjęć w iCloud Photos. Efekt? Zawartość schowka pojawia się w galerii mojego iPhone'a:



Niesamowicie skuteczne!

Zasadniczo zajdziesz tam wszystko czego potrzebujesz, aby tworzyć makra zawierające logikę IF / Switch / For / While / Pause oraz wspomniane ustawianie i przekazywanie zmiennych.

Myślę że najlepszym sposobem na to aby zacząć z Keyboard Maestro, jest zobaczenie mojego kursu na ten temat, dostępnego na eduweb.pl.

Możliwości Keyboard Maestro

Przejdźmy przez to raz jeszcze, ponieważ jest to szczególnie ważne. Keyboard Maestro ma dostęp do wszystkiego, co znajduje się na Twoim komputerze. Plików, urządzeń, zdarzeń, klawiatury a nawet myszki.

Możliwość tworzenia makr, to sposób na to aby łączyć ze sobą wszystkie te możliwości. Co więcej, nic nie stoi na przeszkodzie aby wykorzystać fakt, że potrafisz programować, do tego aby makra sięgały do Twoich skryptów, które wykonują różne operacje (łącznie z połączeniami z zewnętrznym API!).

W ten sposób jesteś w stanie tworzyć automatyzacje, które będą uruchamiać się automatycznie lub będą dostępne pod skrótem klawiszowym. Jeżeli zainstalujesz rozszerzenie do Raycast, możesz też wykonywać makra z jego pomocą.

W rezultacie Keyboard Maestro może stać się centrum automatyzacji i robotów, które będą dla Ciebie pracować! Wystarczy że stopniowo zaczniesz je budować oraz rozszerzać o nowe możliwości. Na początku mogą to być proste transformacje plików, innym razem otwieranie aplikacji a jeszcze innym ... łączenie z zewnętrznymi usługami i

wykonywanie za Ciebie nawet złożonych operacji. W moim przypadku niektóre roboty są w stanie tworzyć dla mnie grafiki bez konieczności otwierania Figmy!

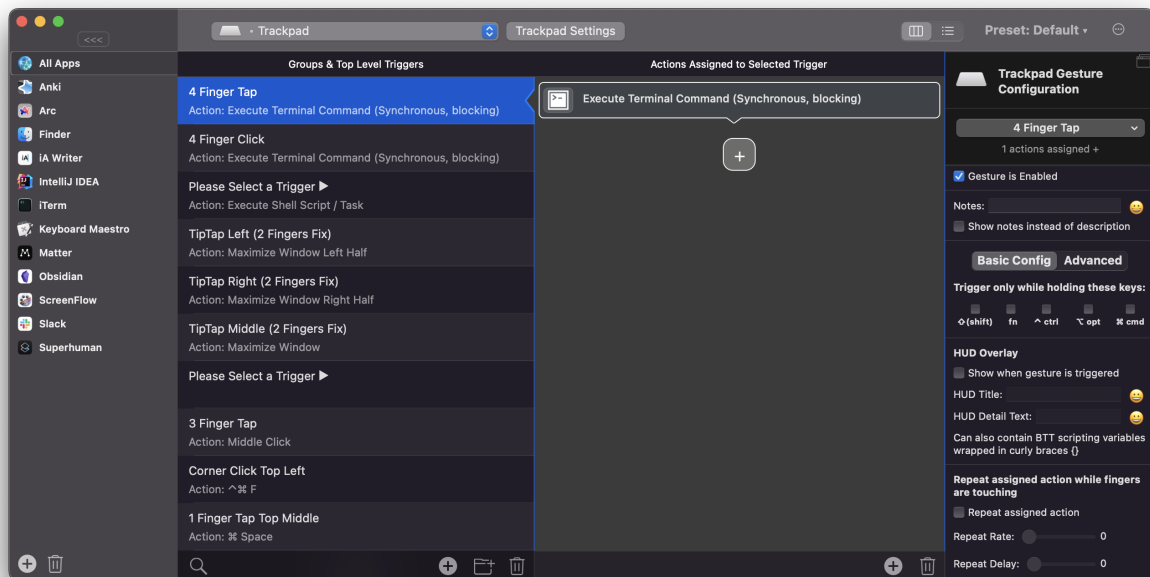
Niewykluczone że na tym etapie, Twoja wyobraźnia podsuwa Ci już kilka pomysłów. Uwierz mi jednak, że wiele z nich przyjdzie Ci, gdy podejmiesz działanie i Raycast oraz Keyboard Maestro staną się dla Ciebie proste w użyciu.

Na tym jednak nie koniec.

Rozszerzenie możliwości trackpada

To jak zaprojektowany jest trackpad w Macbookach, nieustannie robi na mnie wrażenie. Okazuje się jednak, że jego domyślne ustawienia to tylko mały pokaz możliwości. Gdyby tego było mało to ponownie możesz je w pełni wykorzystać, łącząc je z Twoimi programistycznymi umiejętnościami.

Better Touch Tool to aplikacja dostępna w pakiecie Setapp, która działa podobnie jak Keyboard Maestro (trigger -> zestaw akcji), jednak w jej przypadku mamy możliwość definiowania bardzo zaawansowanych gestów.

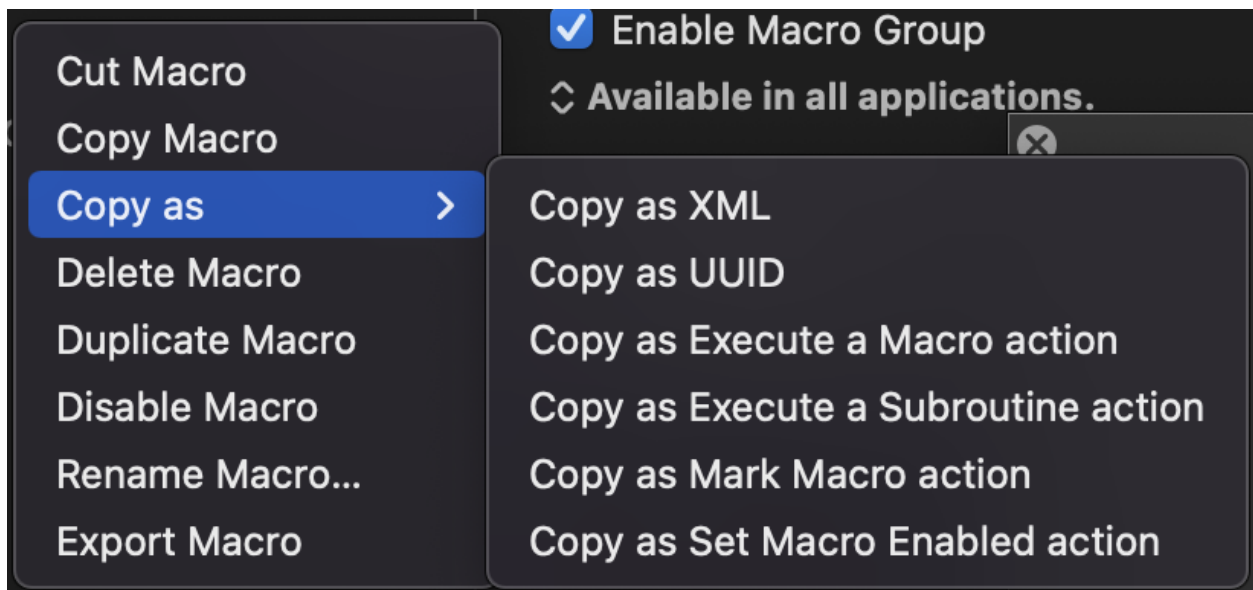


W dodatku na powyższym Screenie widać coś istotnego - możliwość wykonywania skryptów Shell. Wiesz co to oznacza?

Mam dla Ciebie szczególny skrypt, który z pewnością przypadnie Ci do gustu. Efektem jego wykonania jest ... uruchomienie Makra Keyboard Maestro!

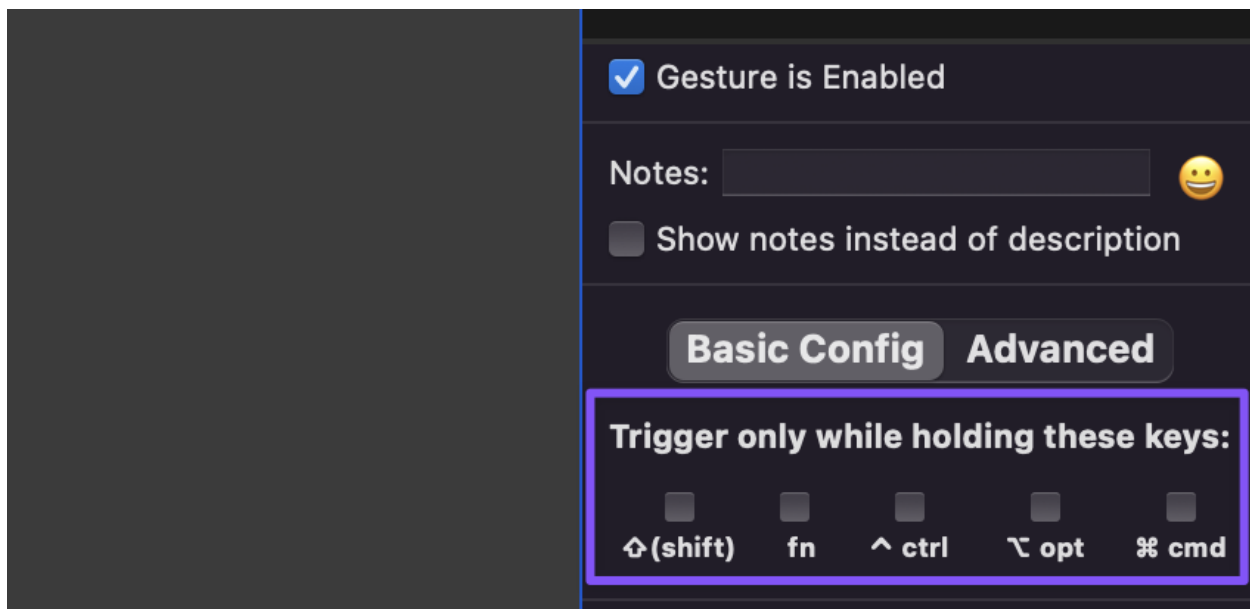
```
osascript -e 'tell application "Keyboard Maestro Engine" to do script "MACRO_UUID"'
```

Wystarczy że w miejscu MACRO_UUID wstawisz wartość, którą możesz skopiować z Keyboard Maestro. Kliknij prawym przyciskiem myszy na makro i wybierz opcję **Copy as** → **Copy as UUID**.



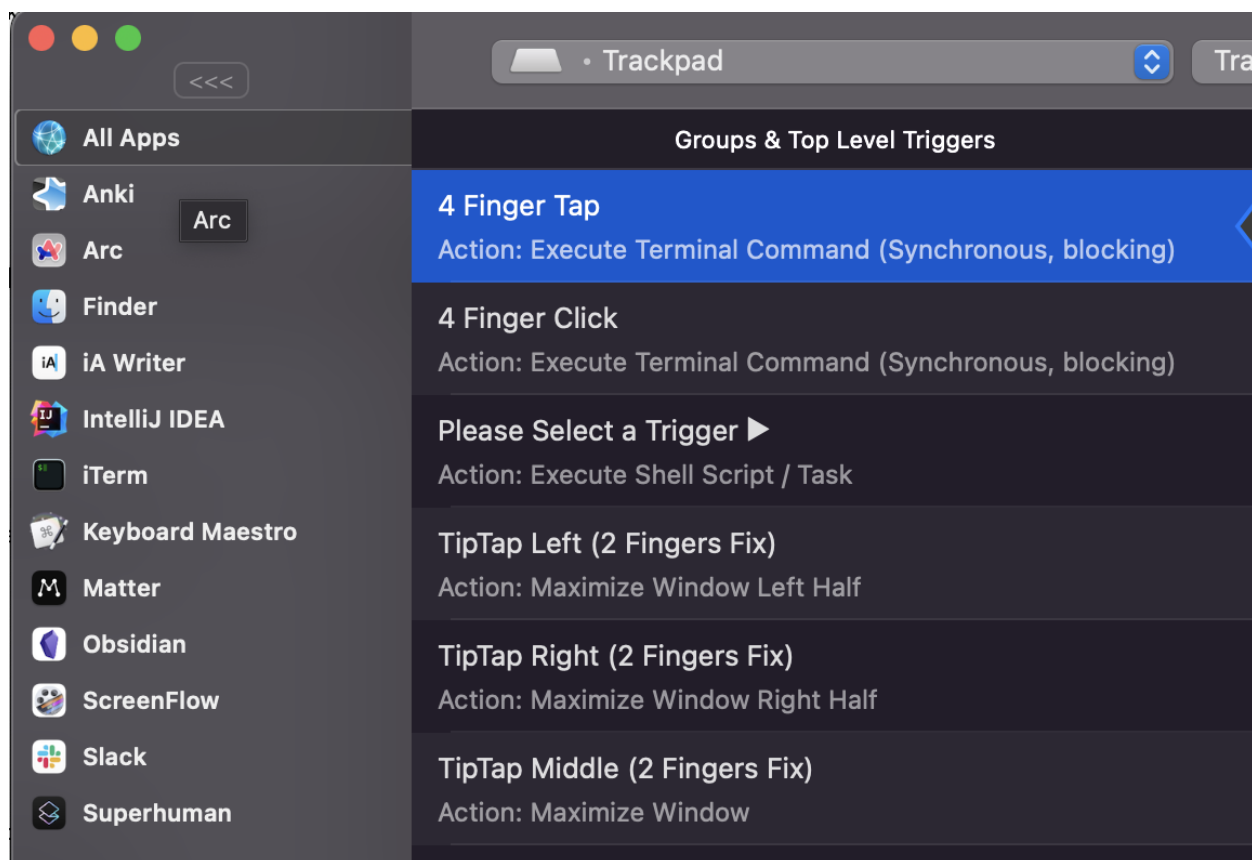
Efekt? Do dowolnego makro Keyboard Maestro (i wszystkich wykonywanych w nim akcji), możesz przypisać dowolny gest wykonany na trackpadzie!

Better Touch Tool to świetna aplikacja, jednak niektóre gesty można uruchomić przypadkowo. Dlatego warto rozważyć uruchomienie ich w połączeniu z dodatkowym klawiszem funkcyjnym. Możesz to skonfigurować na etapie tworzenia makra Better Touch Tool:



W ten sposób unikniesz dodatkowych konfliktów.

Drugim sposobem na unikanie konfliktów, jest wykorzystanie kategorii w postaci Programów w których działają dane gesty. Możesz przypisywać dokładnie te same gesty do różnych akcji, które będą odbierane wyłącznie wtedy, gdy aktywny jest wskazany program.

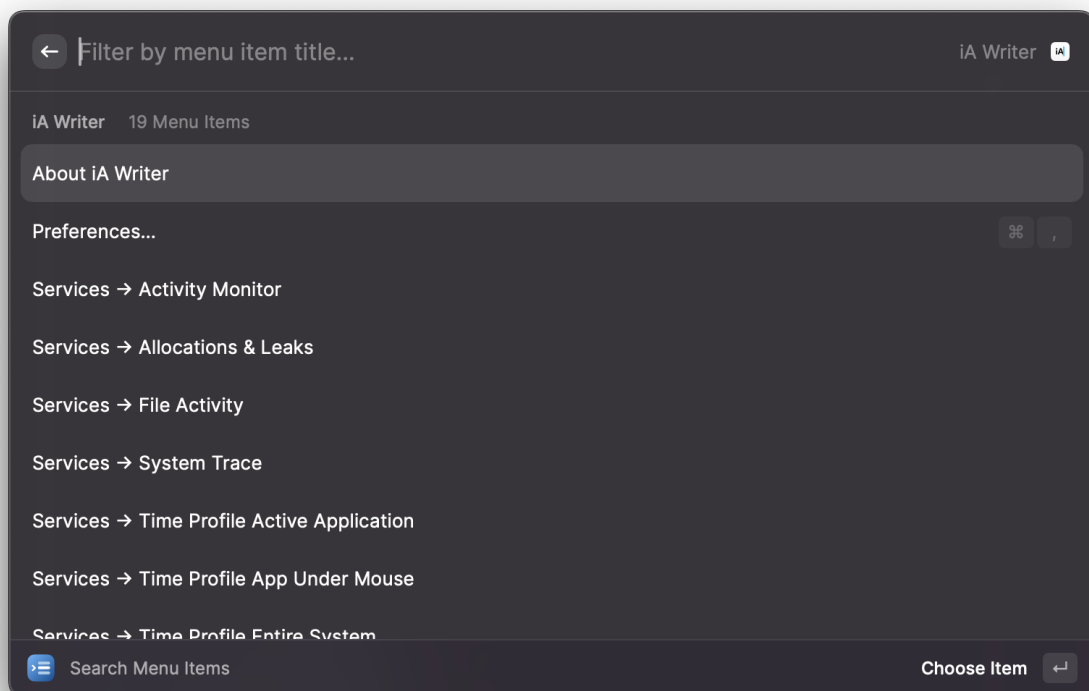


Gdy konfigurowałem makra, wybrałem programy z których korzystam najczęściej, następnie najczęściej wybierane przeze mnie opcje (lub te których uruchomienie sprawiało mi problem) i przypisywałem do nich łatwy do zapamiętania dla mnie gest.

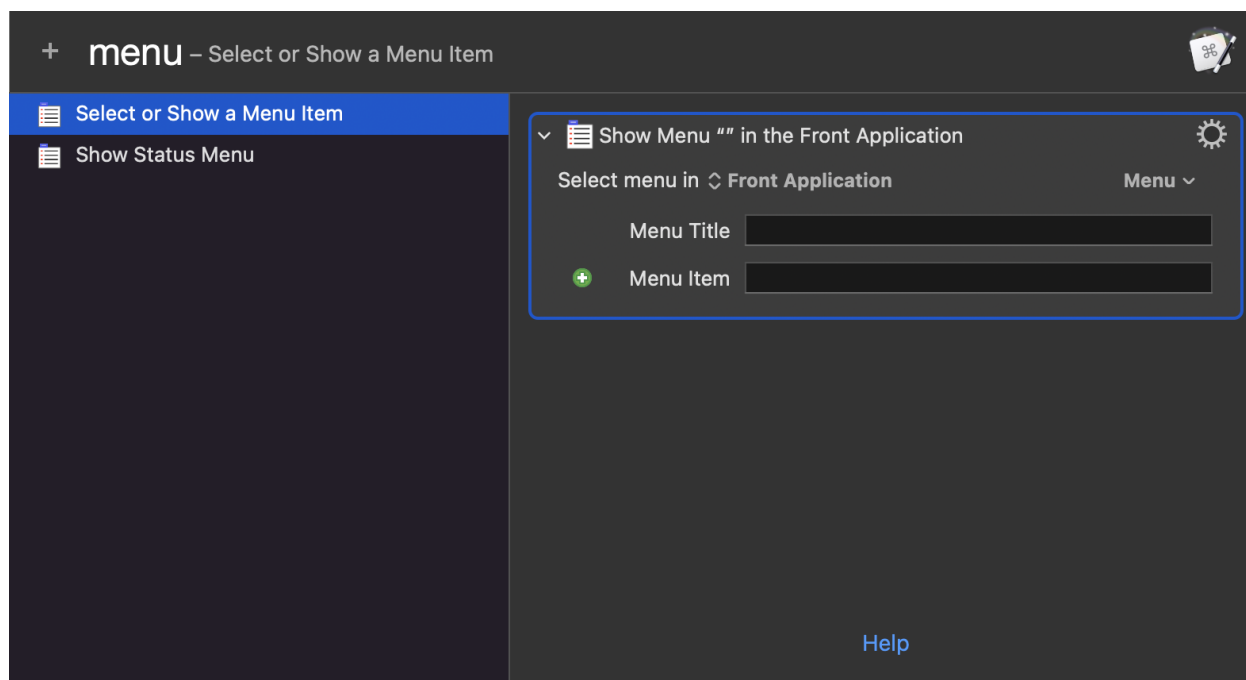
Szybka nawigacja po elementach menu

Raycast posiada genialne rozszerzenie o nazwie `Search Menu Items`. Jego rola jest banalnie prosta - wyświetla listę wszystkich elementów, które aktualnie znajdują się na pasku menu. Dzięki temu nie musisz pamiętać wszystkich skrótów klawiszowych lub co gorsza, ręcznie przeszukiwać opcje które chcesz uruchomić.

Wystarczy, że uruchomisz rozszerzenie i wyszukasz interesujący Cię element. Poza tym jeżeli do danej akcji przypisany jest skrót, również zostanie wyświetlony na liście. Szansa na to, że tym razem go zapamiętasz - rośnie.



Pamiętaj o tym, że korzystanie z **Search Menu Items** nie musi koniecznie odbywać się poprzez akcje podejmowane przez Ciebie. Z pomocą Keyboard Maestro również możesz aktywować konkretne opcje menu:

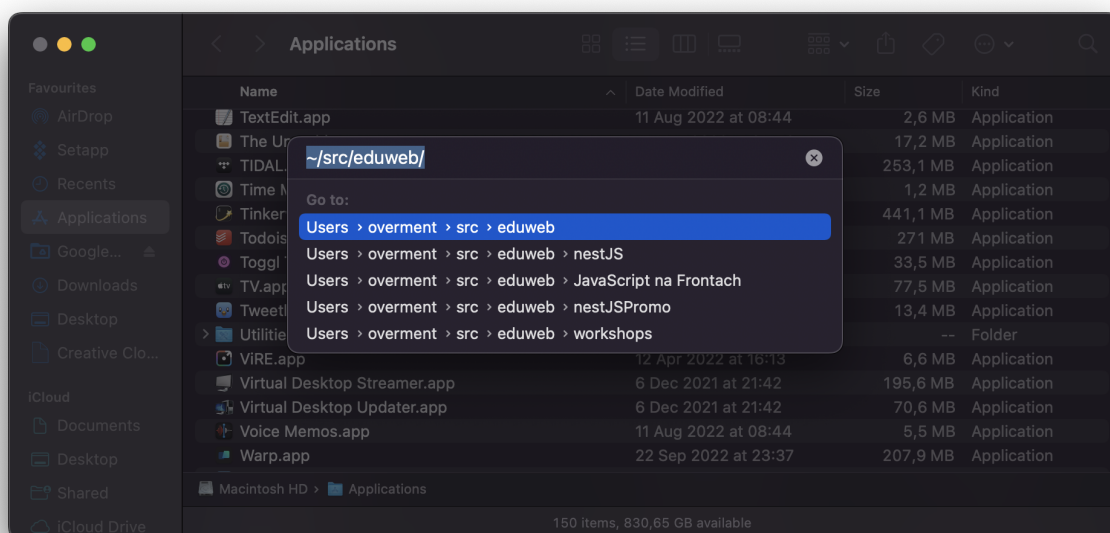


Obie te opcje sprawdzają się najlepiej w przypadku programów, które nie pozwalają na zdefiniowanie skrótów klawiszowych. Tutaj wprost możesz **przypisać własne skróty!**

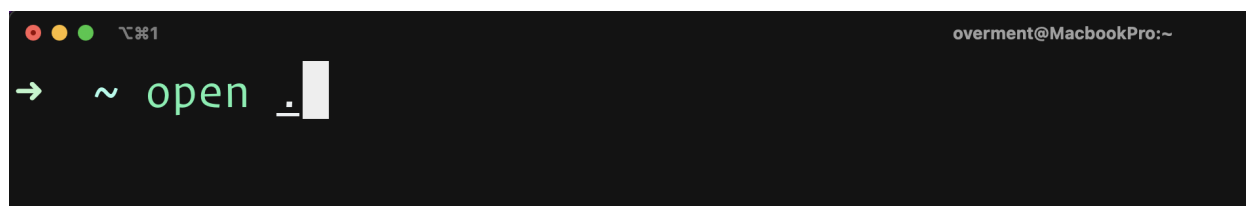
Szybkie otwieranie plików

Finder to prawdopodobnie najsłabszy element całego systemu MacOS i z jakiegoś powodu od dawna nie został naprawiony. Co prawda istnieje kilka alternatyw, natomiast ze względu makr i automatyzacji, sam pozostaję akurat przy Finderze.

Jest tutaj kilka rzeczy na które chciałem zwrócić Twoją uwagę. Przede wszystkim gdy znajdujesz się w dowolnym katalogu, możesz wcisnąć `⌘⇧G` aby wyświetlić okno umożliwiające przeniesienie się do dowolnej ścieżki. Jest to niezwykle użyteczna akcja z pomocą której nawigacja staje się szybsza.



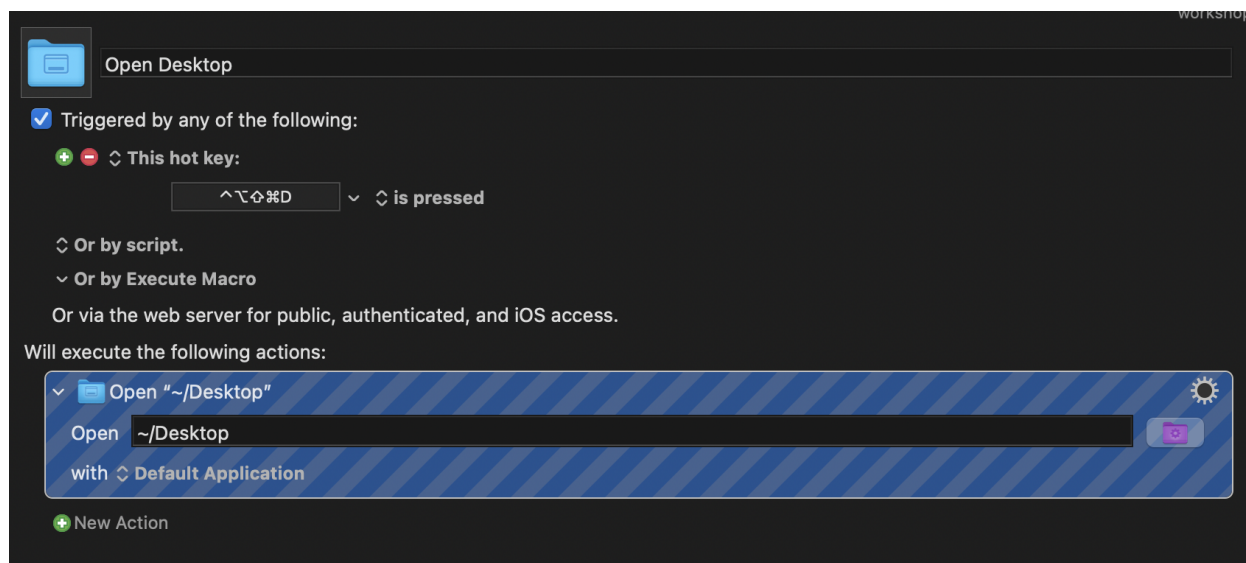
Kolejną świetną sugestią z której korzystam każdego dnia jest polecenie `open`



Wykorzystując je w powyższy sposób, otwierasz bieżący katalog w Finderze.

Ostatnim przykładem jest otwieranie katalogów z poziomu Keyboard Maestro.

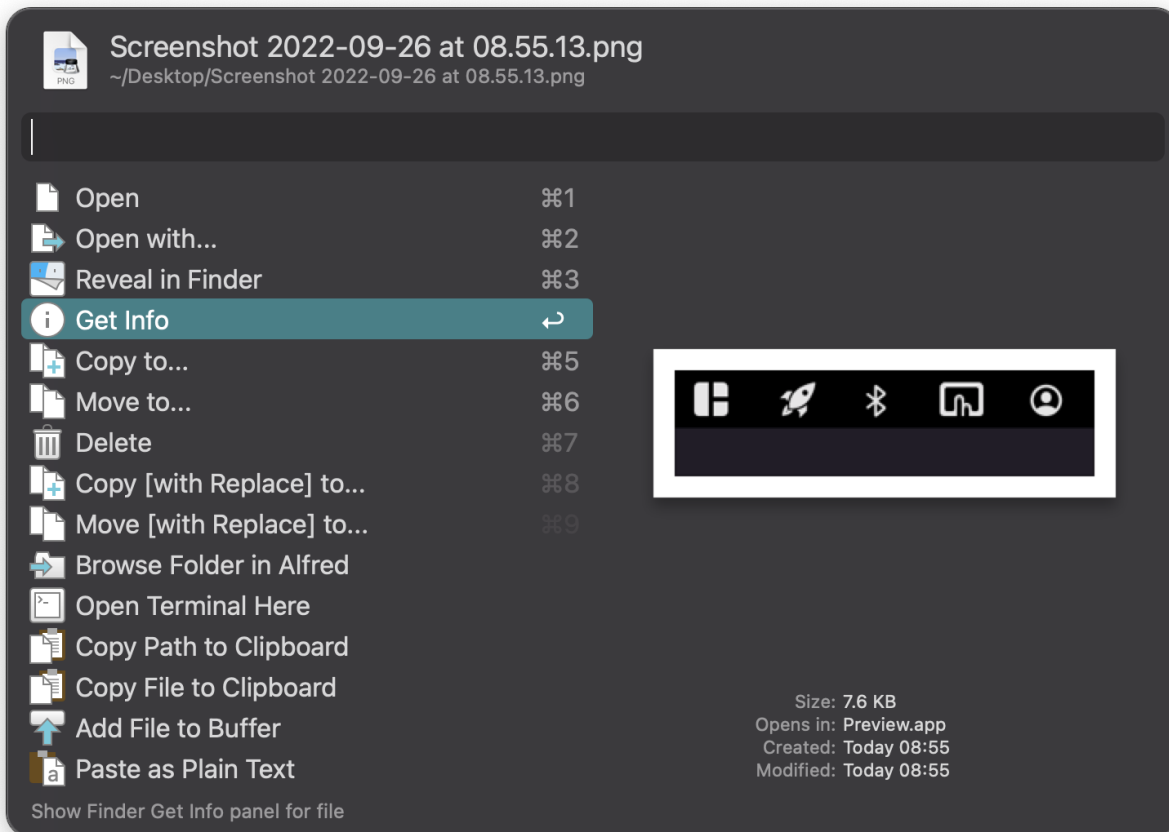
Konkretnie jednym skrótem klawiszowym mogę dostać się do najczęściej odwiedzanych przeze mnie folderów:



Szybkie przenoszenie plików

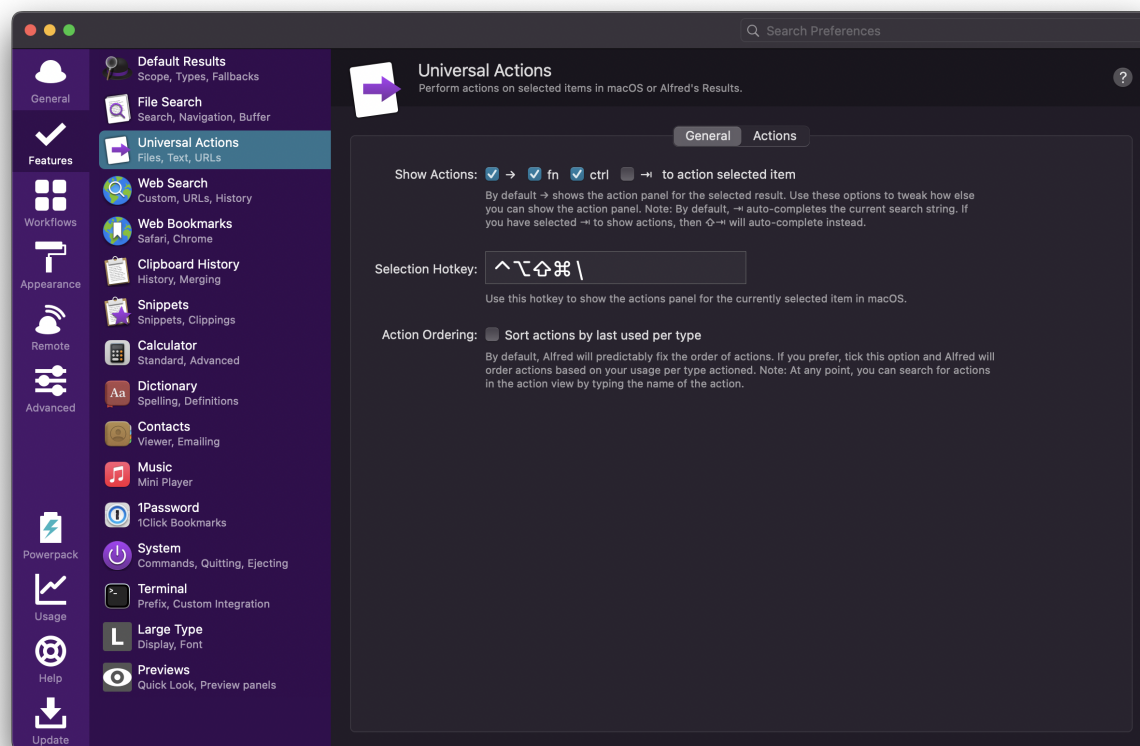
Na ten moment Raycast nie oferuje sensownego sposobu przenoszenia plików. W Finderze również jest to mało wygodne. Z tego powodu nadal w tym jednym celu wykorzystuję Alfred 4 i jego Universal Actions (dostępne w płatnej wersji).

Wygląda to tak:



W praktyce wszystkie zaznaczone pliki w folderze możesz natychmiast przenieść do dowolnego innego katalogu na Twoim komputerze. Alfred charakteryzuje się bardzo szybkim działaniem oraz logicznym sortowaniem plików. W efekcie jest to najlepszy znany mi sposób na szybkie zarządzanie plikami.

Konfiguracja tej opcji jest niezwykle prosta i sprowadza się do przypisania skrótu klawiszowego do Universal Actions w preferencjach Alfreda:

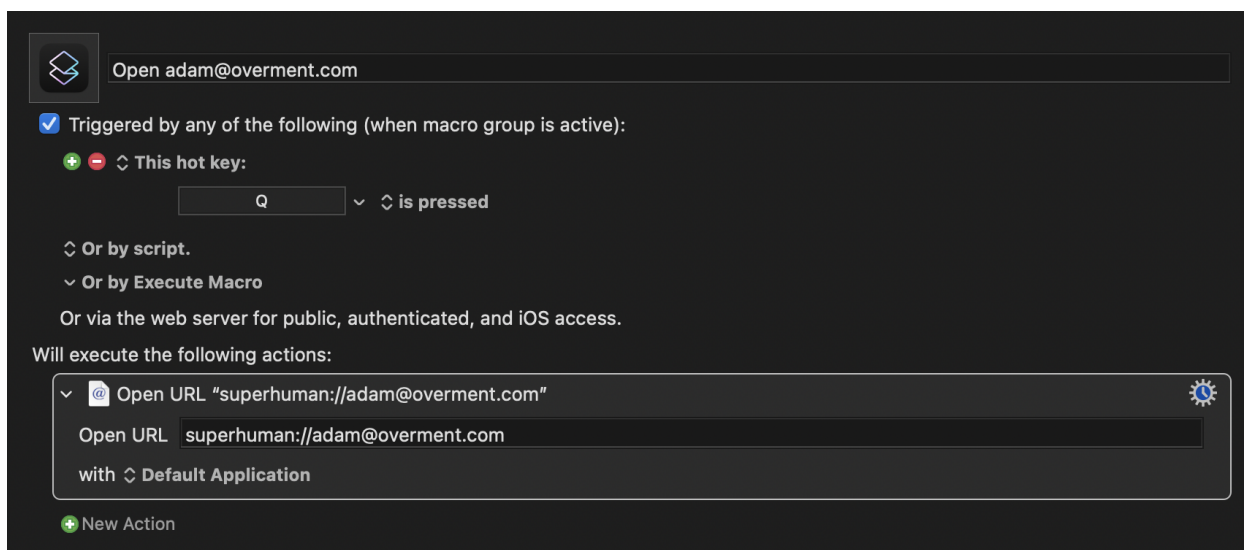


X-Callback-URL

Nie wiem czy wiesz ale w ekosystemie Apple funkcjonują tzw. X-Callback-URL, czyli specjalne linki, które pozwalają wchodzić w interakcję z aplikacjami i ich funkcjami.

Zasada ich wykorzystania polega na uruchomieniu adresu, który zamiast `https` zawiera protokół charakterystyczny dla wybranej aplikacji, np. `superhuman://`.

Jeżeli połączysz ten fakt z możliwością otwierania adresów URL z pomocą Keyboard Maestro, okaże się że ponownie zyskujemy tutaj ciekawe możliwości:

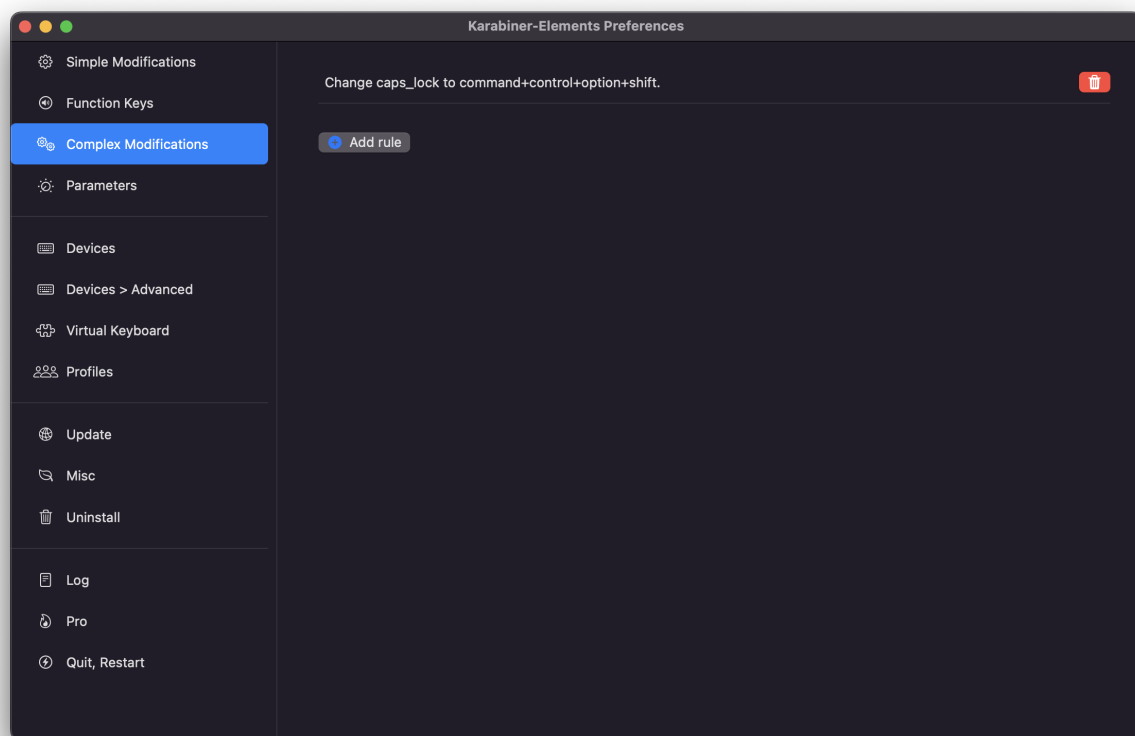


Powyższe makro uruchamia aplikację Superhuman oraz przełącza mnie na moją główną skrzynkę e-mail. Analogicznie mogę tworzyć nowe wiadomości czy przeszukiwać listy.

Wiele aplikacji dostępnych na macOS udostępnia x-callback-url, ale informację na ten temat możesz znaleźć albo w ich dokumentacji albo dopiero po bezpośrednim kontakcie z ich supportem. Najwyraźniej bardzo mało osób jest w stanie wykorzystać to w praktyce. Cóż, Ty przekonasz się że warto to robić.

Hyper Key

Istnieje jeszcze koncepcja tzw. Hyper Key, czyli zmodyfikowania ustawień klawiatury w taki sposób, aby praktycznie dowolny klawisz pełnił rolę klawisza funkcyjnego. Jednym z najlepszych przykładów jest wykorzystanie potencjału klawisza `CapsLock`. Jego pozycja na klawiaturze jest bardzo dobra a jednocześnie jego użyteczność jest zerowa. Z drugiej strony wciśnięcie kombinacji `⌘⇧^↑` jest dość trudne. Dlatego wystarczy zainstalować aplikację Karabiner Elements a następnie w zakładce `Complex Modifications` aktywować wbudowaną regułę `Change caps_lock to command+control+option+shift`



Włączenie tej jednej opcji sprawi że zyskasz dodatkowy klawisz funkcyjny, który możesz przypisać do całej nowej palety skrótów klawiszowych. Na potrzeby Keyboard Maestro lub Raycast - perfekcyjne!

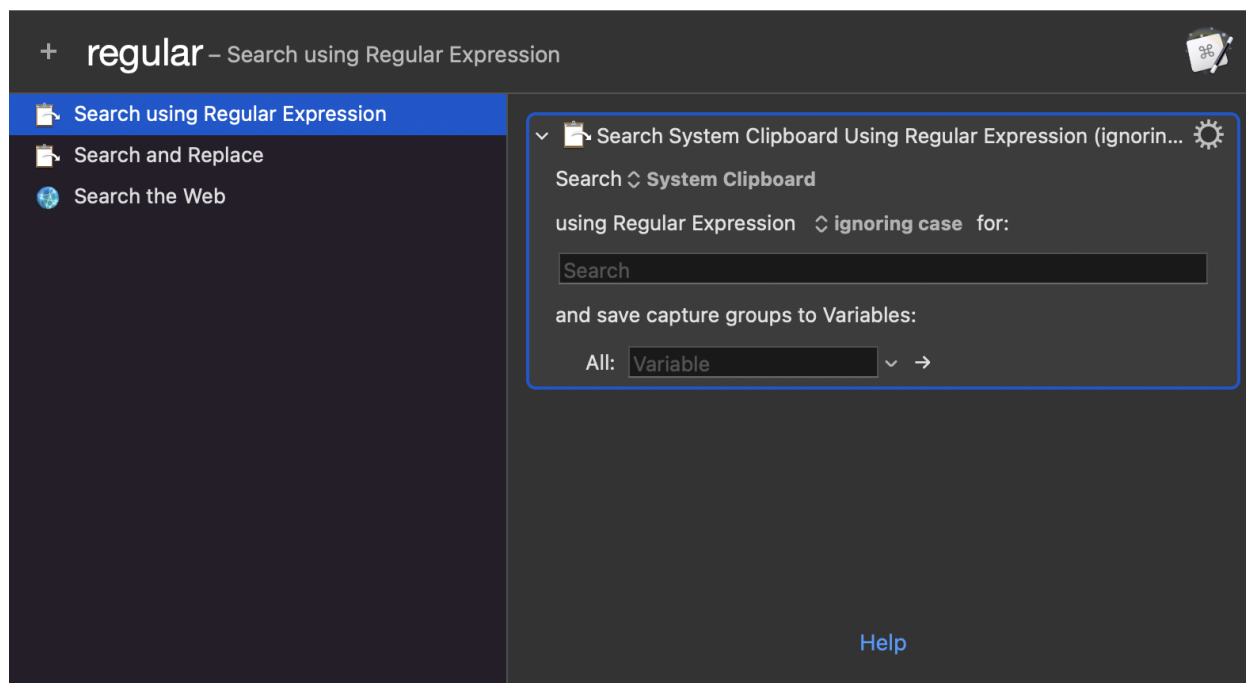
Jeżeli powyższa koncepcja Cię zainteresowała, możesz pójść krok dalej i przeczytać ten artykuł: [Karabiner God Mode](#) w którym Nikita (z którym jestem w stałym kontakcie), opisał szczegółowo jak sam wykorzystuje tę aplikację. Ostrzegam Cię tylko, że to zaawansowany przykład.

Praca z tekstem na poziomie

Pisanie kodu i jego transformacje z pomocą multi-kursora raczej nie kryją przed Tobą tajemnic. Jednak jest szansa, że pisanie wyrażeń regularnych stanowi dla Ciebie pewne wyzwanie. Twoim zadaniem jest to zmienić.

Chodzi o to, że tworzenie automatyzacji oraz makr, na każdym kroku będzie wymagało od Ciebie sprytnych modyfikacji mniejszych i większych fragmentów tekstu i musi odbywać się to w sposób automatyczny.

Jest to szczególnie przydatne, ze względu na to, że **wszystkie** wykorzystywane przez nas techniki, będą wymagać pisania wyrażeń regularnych. Spójrz:



Wyszukiwanie, filtrowanie, dopasowanie, mapowanie - wszystkie te operacje oferują więcej, jeżeli potrafisz napisać wyrażenie, które dopasuje oczekiwany przez Ciebie wzór w podanym tekście.

Dobra wiadomość jest taka, że możesz obejrzeć mój 35 minutowy warsztat aby zrozumieć wyrażenia regularne i przynajmniej móc z nimi wygodnie pracować.

[Zobacz warsztat na eduweb.pl](http://eduweb.pl)

Potem przeniesienie tej wiedzy do Keyboard Maestro nie będzie stanowiło już dla Ciebie problemu. Tymczasem warto jeszcze dodać, że aplikacje Expressions (dostępna w Setapp) również ułatwi Ci pisanie i testowanie wyrażeń.

Poza Keyboard Maestro czy Twoimi skryptami, wyrażenia regularne sprawdzą Ci się w zwykłych edytorach tekstowych z których niemal każdy udostępnia opcję "Znajdź i Zamień", która wspiera regexp.

Nie będę Cię dłużej przekonywał. Ta umiejętność jest niezbędna dla Ciebie i Twojej dalszej przygody z automatyzacją i usprawnieniami swojej pracy. A jej zdobycie zajmie Ci 35 minut podczas mojego warsztatu. To serio jest proste.

Automatyczna transformacja tekstu

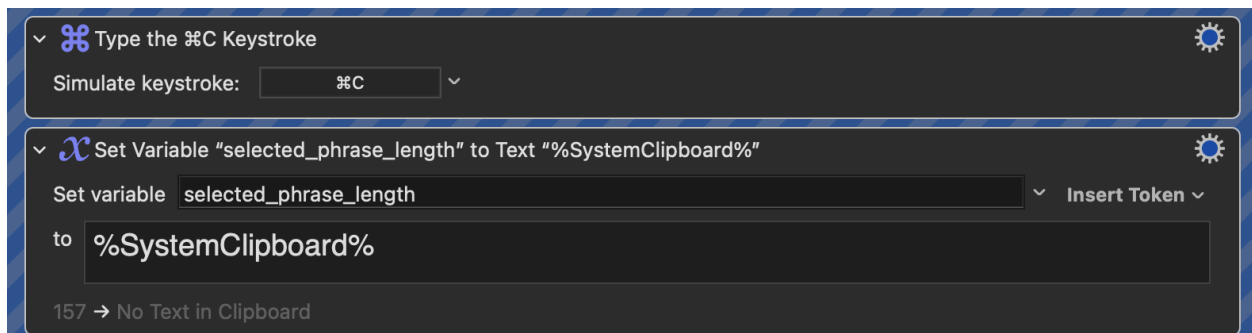
Okazuje się, że sama transformacja tekstu może iść nieco dalej, niż mogłoby się początkowo wydawać. Z pomocą Rayacst lub Keyboard Maestro możesz przygotować Komendy i Makra, które zamienią zaznaczony tekst.

Wspominałem Ci o usłudze Deepl. W moim przypadku wystarczy, że zaznaczę tekst i wcisnę skrót klawiszowy aby go przetłumaczyć. Zrobię to teraz z tym akapitem.

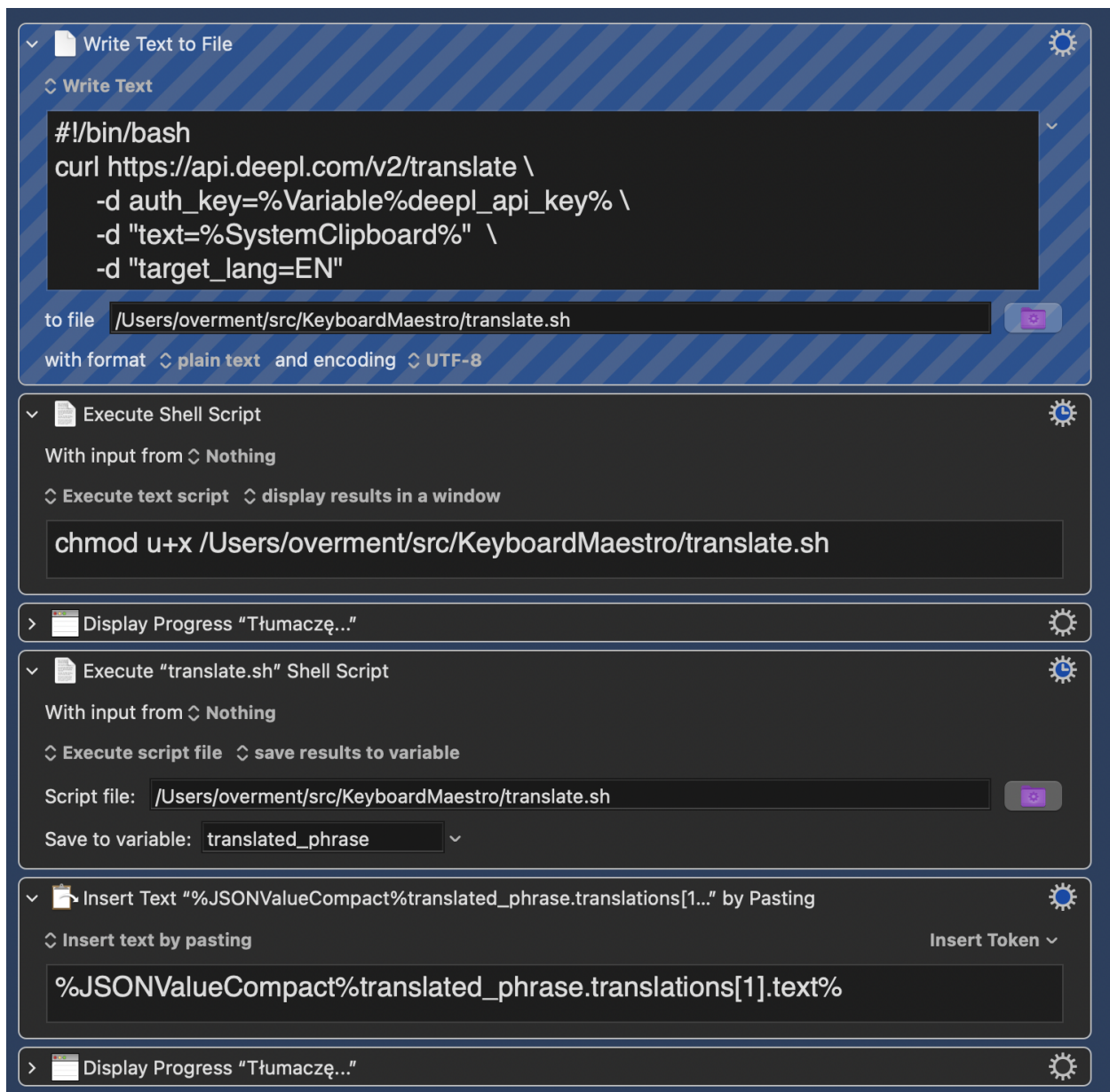
I mentioned to you about Deepl service. In my case, all I have to do is select the text and press a keyboard shortcut to translate it. I will do it now with this paragraph.

Jak widzisz jakość tłumaczenia może nie jest perfekcyjna, ale wystarczająca. Tym bardziej że usługi takie jak Deepl nieustannie się rozwijają.

Przygotowanie takiego makra wymagało ode mnie stworzenia prostego CURLa, który przesyła zaznaczony tekst do API Deepl. W pierwszym kroku symuluję wciśnięcie skrótu ⌘C aby skopiować tekst:



A potem przekazuję go do wspomnianego skryptu i odczytuję odpowiedź:



Do składni Keyboard Maestro, szczególnie tej dotyczącej obiektów JSON trzeba przywyknąć. Ostatecznie jednak jest to stosunkowo proste.

Na podobnej zasadzie utworzyłem makra, które zamieniają mi Markdown na HTML czy Markdown na Styled Text (z pomocą narzędzia [pandoc](#) i jego CLI). Po co? Ano po to, że teraz mogę pisać wiadomości e-mail w edytorze [iA Writer](#) a potem z pomocą makra przenieść je wprost do [Superhuman](#).

Nic nie stoi na przeszkodzie aby wykorzystywać tutaj bardziej zaawansowane transformacje tekstu. Albo z pomocą własnych skryptów albo narzędzi CLI takich jak

choćby wspomniany pandoc.

JavaScript jest wszędzie

Myślę że jasno widzisz już jak potężne możliwości daje JavaScript, szczególnie uwzględniając Node.js oraz Frameworki takie jak React, Vue czy Svelte. Znajomość tego ekosystemu może być wykorzystana nie tylko we wspomnianych przeze mnie wyżej aplikacjach. Przykładów jest znacznie więcej i każdy z nich charakteryzuje to, że może ułatwiać Ci pracę i otwierać możliwości niedostępne dla osób, które nie potrafią programować.

Co ciekawe JavaScript a nawet TypeScript pojawiają się nie tylko wewnątrz aplikacji desktopowych czy w kontekście tworzenia Twoich własnych skryptów. Jest on obecny chociażby w takich usługach jak Retool czy Airtable. W takiej sytuacji fakt, że znasz JavaScript, pozwala Ci korzystać z ich najbardziej zaawansowanych opcji.

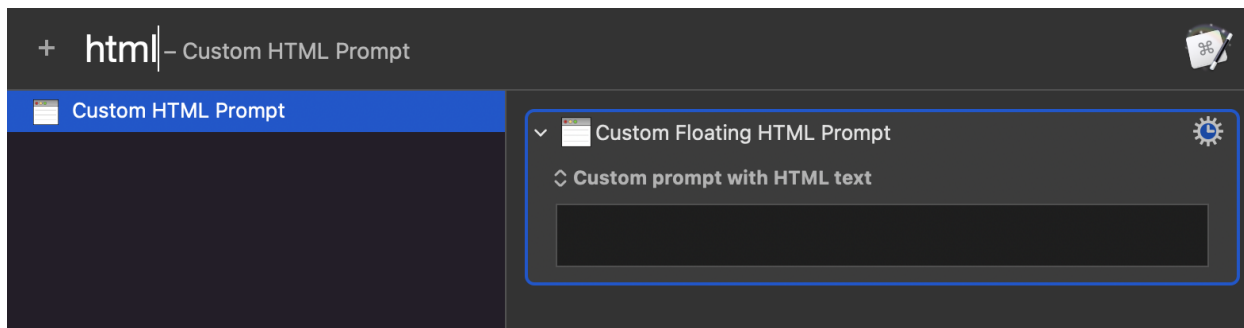
Scenariusz jest tutaj bardzo prosty - jeżeli ekosystem JavaScriptu (JS, TS, Node.js, frameworki) i związany z nim tooling (np. Vite, pnpm czy bun) nie są Ci znane, to całkiem dobrym pomysłem będzie rozwój tych umiejętności.

Tutaj jak zwykle przydatne materiały znajdziesz na eduweb.pl oraz egghead.io.

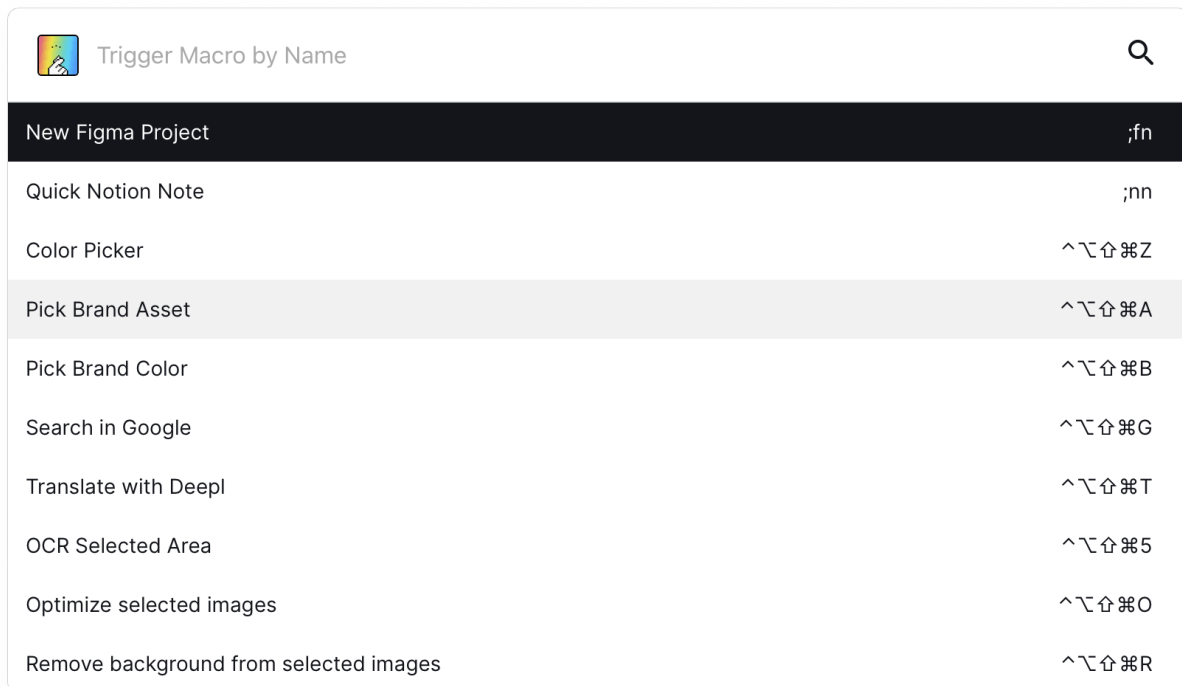
Tymczasem pokażę Ci jeden z najbardziej zaawansowanych przykładów wykorzystujących JavaScript na potrzeby automatyzacji, jaki do tej pory stworzyłem. Poznaj designmaestro.io!

Własne rozszerzenia i ogromne możliwości

W chwili gdy pracowałem nad rozszerzeniem Confetti o którym wspominałem, odkryłem jedną z funkcji Keyboard Maestro, która umożliwia wyświetlenie okna z dowolnym kodem HTML.



W praktyce funkcja ta korzystając z Chromium, wyświetla okienko przeglądarki, które może być całkowicie przeźroczyste i wyświetlać nasz kod HTML, CSS oraz ... JavaScript! W rezultacie możesz zakodować niemal dowolny interfejs i wyświetlić go z pomocą Keyboard Maestro. Dodanie do tego interakcji stworzonych z pomocą JavaScriptu dało początek Design Maestro:

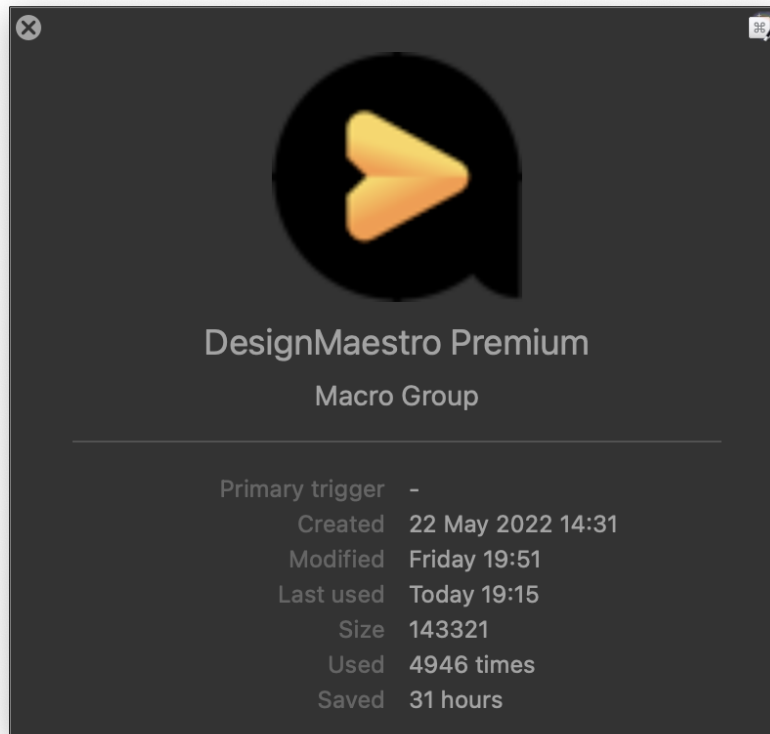


Jak łatwo się domyślić, przygotowanie takiego projektu było dość wymagające zarówno z punktu widzenia jego objętości jak i złożoności logiki. Uwzględniłem tutaj zarówno obsługę zmiennych, skrótów klawiszowych jak i interakcję ze schowkiem a nawet z webhookami. Dlatego zdecydowałem się na utworzenie całości z pomocą Svelte oraz Vite. Dzięki takiej konfiguracji rezultatem końcowym był czysty kod JavaScript osadzony inline'owo w kodzie HTML, który trafił bezpośrednio do Keyboard Maestro.

W tym projekcie był jeszcze jeden istotny element, który zawsze biorę pod uwagę tworząc automatyzacje oraz makra. Mowa o możliwości ponownego wykorzystania oraz elastycznego rozbudowania. Design Maestro zaprojektowałem tak, aby interfejs formularzy i list był generowany na podstawie spójnego obiektu JSON. Dzięki temu w tym momencie zbudowanie kolejnego makra opartego o Design Maestro, sprowadza

się do przygotowania jednego obiektu JSON i skorzystania z funkcji, która renderuje widok. Proste!

Ten element reużywalności jest tutaj bardzo ważny, ponieważ raz zainwestowany czas zwraca mi się teraz wielokrotnie. I to dosłownie. Spójrz:







DesignMaestro

Macro Group

Primary trigger -

Created	17 May 2022 03:50
Modified	8 Sep 2022 22:31
Last used	Today 17:57
Size	152508
Used	2493 times
Saved	57 hours





DesignMaestro Priv

Macro Group

Primary trigger -

Created	29 May 2022 21:24
Modified	Today 19:03
Last used	Today 19:04
Size	52859
Used	2284 times
Saved	54 hours

Design Maestro od maja (do października) zaoszczędził mi 142 godziny. To niemal miesiąc etatowej pracy a to tylko makra z grupy Design Maestro. Wszystkich wykorzystywanych przeze mnie jest przynajmniej kilka razy więcej.

To wszystko potwierdza Ci jedno: warto inwestować czas na aktywności, które będą Ci go zwracać wielokrotnie i to na przestrzeni kolejnych lat. I wiesz co jest najlepsze? Możesz teraz przejść na <https://www.designmaestro.io>, pobrać paczkę **za darmo** i na podstawie przygotowanej przeze mnie dokumentacji, wykorzystać efekt mojej pracy.

Opis tej dokumentacji znajdziesz tutaj:



oraz w materiałach udostępnianych po odebraniu lub zakupie (jest opcja Pay What You Want).

Od siebie dodam, że nie musisz tego robić, ponieważ bardzo podobny interfejs znajdziesz teraz w Raycast. Aczkolwiek muszę przyznać, że na ten moment wygodniej tworzy się makra w Keyboard Maestro, ponieważ tam pisanie kodu jest opcjonalne i wiele opcji jest dostępna "od razu". W przypadku Raycast niemal wszystko musisz pisać od podstaw.

Twoje mikroserwisy

Myślę że jasne już jest dla Ciebie, że jesteś w stanie tworzyć makra i automatyzacje, które będą wykonywać operacje na Twoim komputerze. Z pomocą własnych skryptów oraz narzędzi CLI, możesz też komunikować się z zewnętrznymi usługami. To wszystko może jednak pójść krok dalej, ponieważ Twoje umiejętności wcale się tutaj nie kończą.

Jeżeli znasz JavaScript, możesz tworzyć własne aplikacje, które mogą działać na Twoim serwerze i komunikować się z Twoim komputerem oraz zewnętrznymi usługami. One same mogą udostępniać API, które możesz wykorzystywać na różne sposoby, które za chwilę Ci pokażę.

Przede wszystkim jednak, potrzebujesz swojego miejsca w sieci. Całkiem prawdopodobne, że posiadasz już swój hosting. Tutaj jednak potrzebujesz czegoś więcej - własnego serwera, np. VPS.

VPS

Sam posiadam VPS wykupiony na DigitalOcean. Jeżeli jednak potrzebujesz VPS do nauki i małych aplikacji, niewykluczone że najlepszym wyborem będzie dla Ciebie <https://mikr.us> od Jakuba Mrugalskiego.

W każdym razie kluczowe jest tutaj to, aby mieć możliwie swobodny dostęp do tej maszyny i móc konfigurować na niej praktycznie dowolne usługi.

Na Digital Ocean skonfigurowanie Dropleta jest bardzo proste. Jeżeli chodzi o system operacyjny, najnowsza wersja Ubuntu będzie najlepszym wyborem. Następnie konieczne będzie wybranie sposobu uwierzytelnienia połączenia. W naszym przypadku może być to klucz SSH, który konfigurowaliśmy wcześniej:

Authentication ?

☒ **SSH keys**
A more secure authentication method

☐ **Password**
Create a root password to access Droplet (less secure)

Choose your SSH keys  Select at least one key.

☐ Select all ☐ JS

[New SSH Key](#)

Możesz tutaj utworzyć nowy klucz SSH i ustawić go na zawartość pliku `~/.ssh/id_rsa.pub`. Aby łatwo pobrać jego zawartość uruchom polecenie:

```
cat ~/.ssh/id_rsa.pub | pbcopy
```

Na przygotowanie serwera potrzebujesz maksymalnie kilka minut. Aby się z nim połączyć uruchom polecenie

```
ssh root@ADRES_IP
```

Gdzie naturalnie adres IP wskazuje na Twój droplet. Jeżeli wszystko pójdzie ok, połączenie powinno zostać nawiązane. Pamiętaj, że robiąc to za pierwszym razem, musisz wpisać `yes` w pytaniu, które się pojawi.

```
Memory usage: 24%          IPv4 address for eth0: 10.19.0.10
Swap usage:    0%          IPv4 address for eth1: 10.135.28.158
Processes:    96

0 updates can be applied immediately.

Last login: Sun Sep 25 19:24:50 2022 from 87.239.222.143
root@overment:~# #
```

Dalsza część konfiguracji wymaga przede wszystkim:

- ustawienia serwera HTTP (np. NGINX) i pełny tutorial znajdziesz tutaj:
<https://www.digitalocean.com/community/tutorials/how-to-install-nginx-on-ubuntu-22-04>
- ustawienia domeny oraz SSL i pełny tutorial znajdziesz tutaj:
<https://www.digitalocean.com/community/tutorials/how-to-secure-nginx-with-let-s-encrypt-on-ubuntu-20-04>
- ustawienia aplikacji node.js i pełny tutorial znajdziesz tutaj:
<https://www.digitalocean.com/community/tutorials/how-to-set-up-a-node-js-application-for-production-on-ubuntu-20-04>

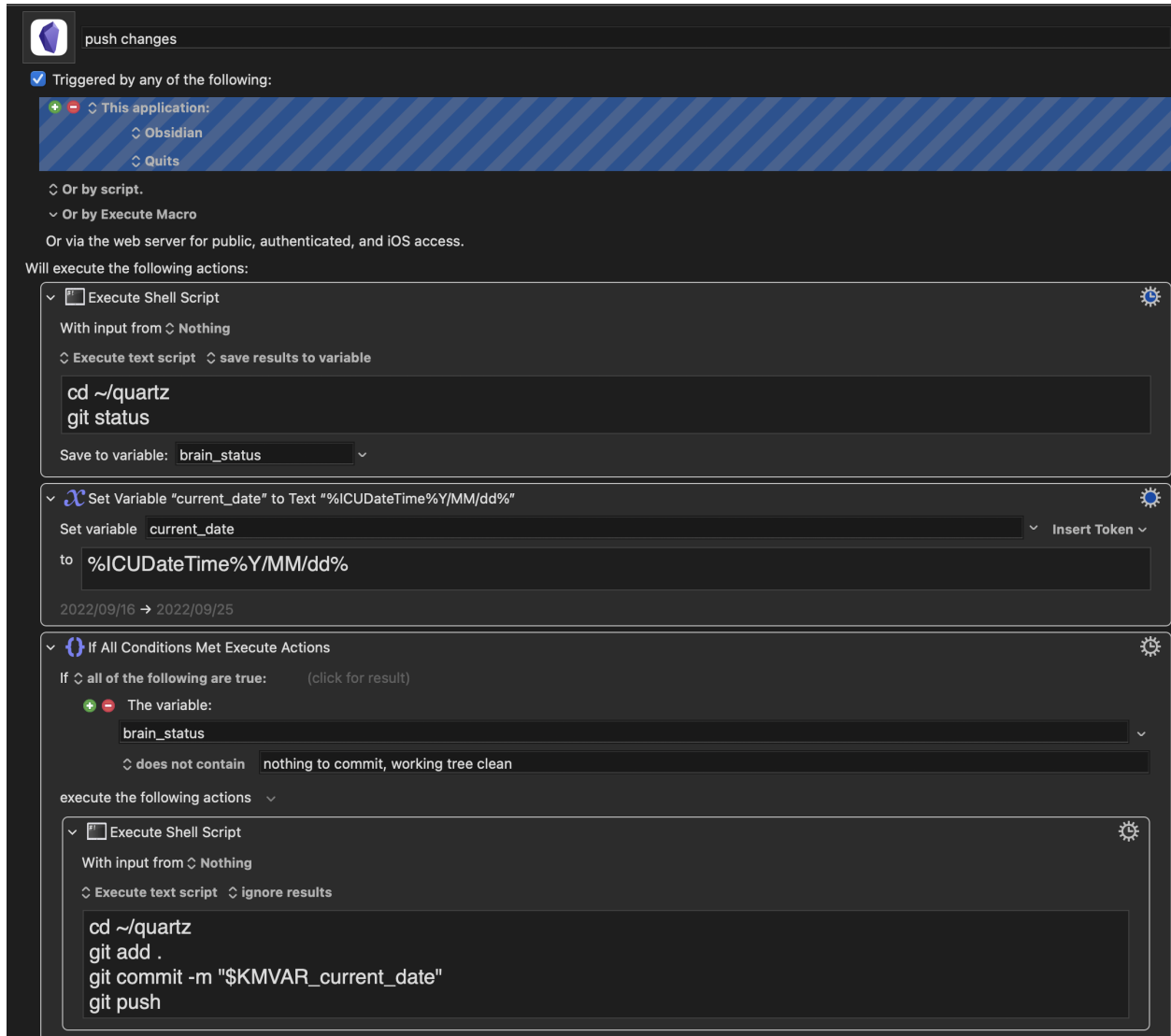
Przejsście przez powyższe artykuły jest niesamowicie wartościowe pod kątem rozwoju Twoich automatyzacji oraz szeroko pojętych programistycznych umiejętności. W dodatku niemal za każdym razem gdy będziesz potrzebować konfiguracji jakiejś usługi na VPS, na blogu Digital Ocean znajdziesz bardzo wartościowe materiały, które krok po kroku Cię przez nią przeprowadzą.

Ważne: w przypadku DigitalOcean opłaty naliczane są za czas w którym Droplet jest aktywny. Jedynym sposobem na powstrzymanie naliczania opłat jest usunięcie Dropleta (opcja Destroy).

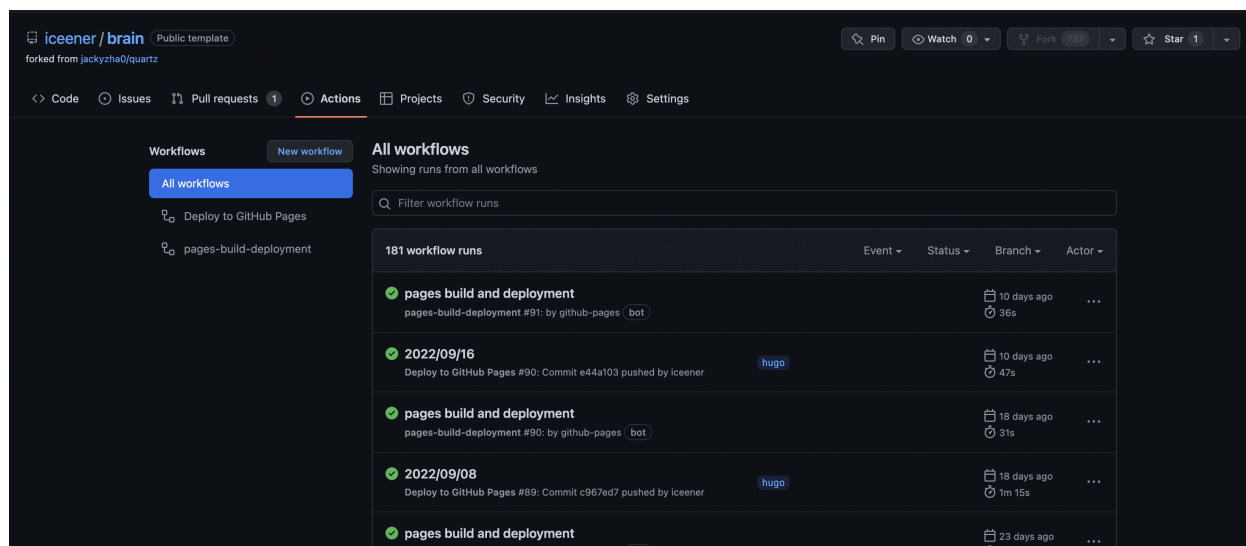
Poza tym w kontekście konfiguracji Twojego serwera i aplikacji, które na nim działają, najlepszym pomysłem jest wykorzystanie usługi Github Actions. Efekt jaki Cię interesuje polega na tym, aby w momencie gdy na branch main Twojej aplikacji trafią zmiany, proces deploymentu odbędzie się automatycznie.

W przypadku mojego projektu brain.overment.com, scenariusz jest jeszcze prostszy. Notatki i tym samym zawartość tej strony rozwijam bezpośrednio w Obsidianie. Moment wyjścia z tej aplikacji jest przechwytywany jako trigger w Keyboard Maestro. Makro

wykonuje kilka poleceń terminala, dotyczących zatwierdzenia zmian i dodania ich do repozytorium. Co ciekawe dzieje się to tylko wtedy, gdy faktycznie jakieś zmiany zostały dokonane:



A potem w momencie gdy zmiany faktycznie pojawią się w zdalnym repozytorium, wykonywana jest seria akcji w Github Actions, odpowiedzialnych za zbudowanie i publikację aplikacji.



Także jak widzisz, w przypadku tego konkretnego projektu, nie mam jakiegokolwiek potrzeby nawet zaglądania na chwilę na serwer czy do repozytorium. Wszystko dzieje się absolutnie bez mojego udziału. 100% uwagi skupia się wyłącznie na tworzeniu treści.

W kontekście własnego VPS, najbardziej będą interesować Cię tematy takie jak:

- obsługa terminala
- konfiguracja NGINX (zajrzyj tutaj: <https://wiedza.mrugalski.pl/product/kurs-konfiguracji-nginx>)
- konfiguracja domen i SSL (np. bezpłatny Let's Encrypt)
- konfiguracja podstawowych usług (bazy danych, firewall)
- konfiguracja aplikacji Node.js (narzędzie pm2)
- CI&CD (Github Actions)

Informacje na praktycznie każdy z powyższych tematów znajdziesz na blogu Digital Ocean. Nawet podstawowa wiedza na ten temat wystarczy Ci do budowania swoich małych usług.

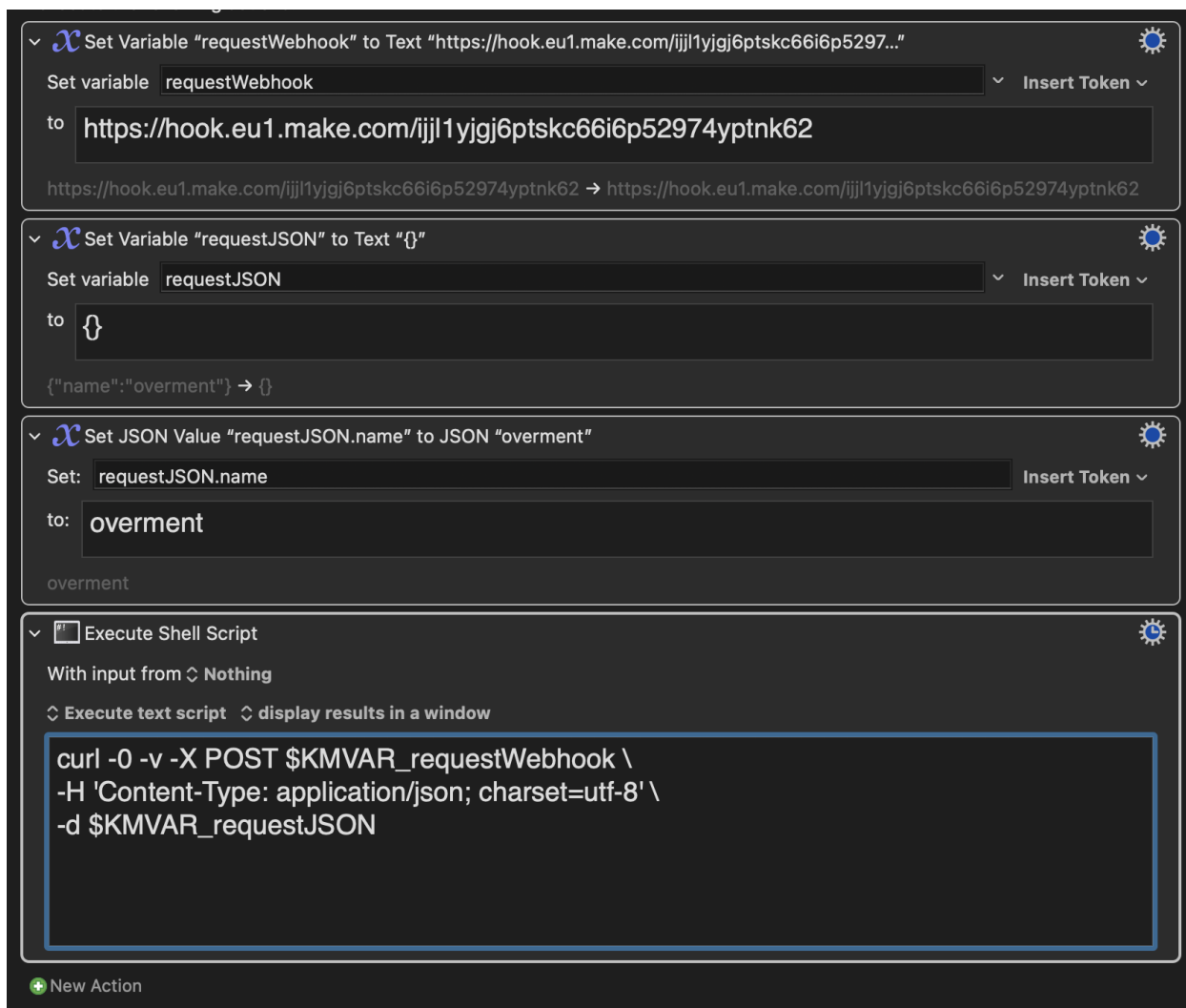
Połączenie Keyboard Maestro z Twoim API

Gdy będziesz mieć już działającą usługę i skonfigurowany VPS, kolejnym krokiem będzie podłączenie się do niej poprzez np. Makra Keyboard Maestro lub rozszerzenia Raycast.

Masz tutaj przynajmniej kilka opcji:

- wykorzystanie `curl` lub `wget` do utworzenia zapytania HTTPS do Twojego API
- wykorzystanie opcji Design Maestro, umożliwiającej przesłanie dowolnych danych na wskazany na Ciebie adres

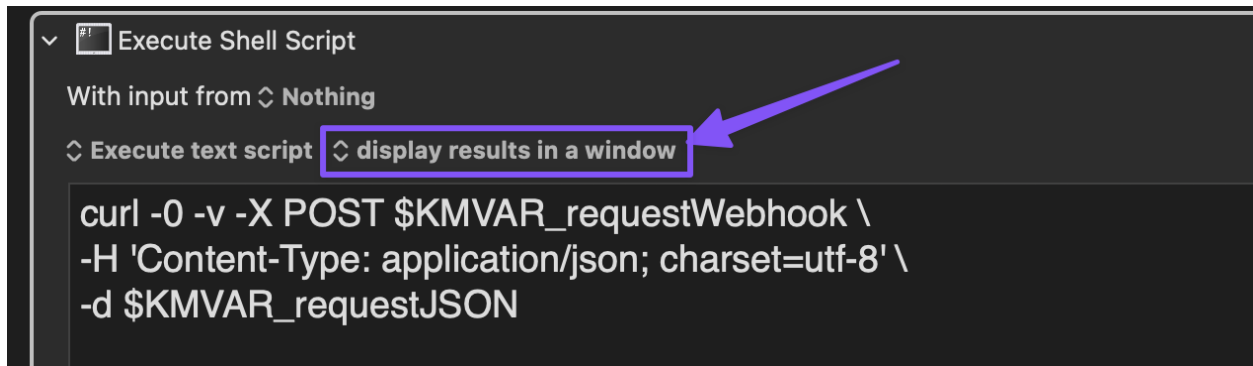
Pierwszy przykład wygląda tak polega na tym, aby przygotować obiekt JSON a następnie przesłać go na wskazany adres. W moim przypadku jest to tymczasowy webhook:



Powyższe makro możesz pobrać [tutaj](#).

Wystarczy że podmienisz zawartość adresu z webhookiem oraz naturalnie dane, które chcesz przesłać na webhook. Warto dodać, że rezultat tego zapytania również może

trafić do zmiennej (opcja Save results to variable):



W porządku! W tym momencie wiesz już niemal wszystko, co związane jest z moją codziennością. Chciałbym jednak pójść z tym o jeszcze jeden krok. Pomimo nawet bardzo wysokiej optymalizacji Twojej pracy oraz wysokich programistycznych kompetencji, nadal przygotowanie własnych aplikacji może zajmować Ci dużo czasu. Tym bardziej że często mówimy o bardzo prostych zadaniach, takich jak np. przesłanie informacji z jednego systemu do drugiego czy wysłanie prostego maila.

Z tego powodu Twoja uwaga, powinna zostać skierowana jeszcze w jedno miejsce - platformę [Make.com](https://make.com).

Make.com

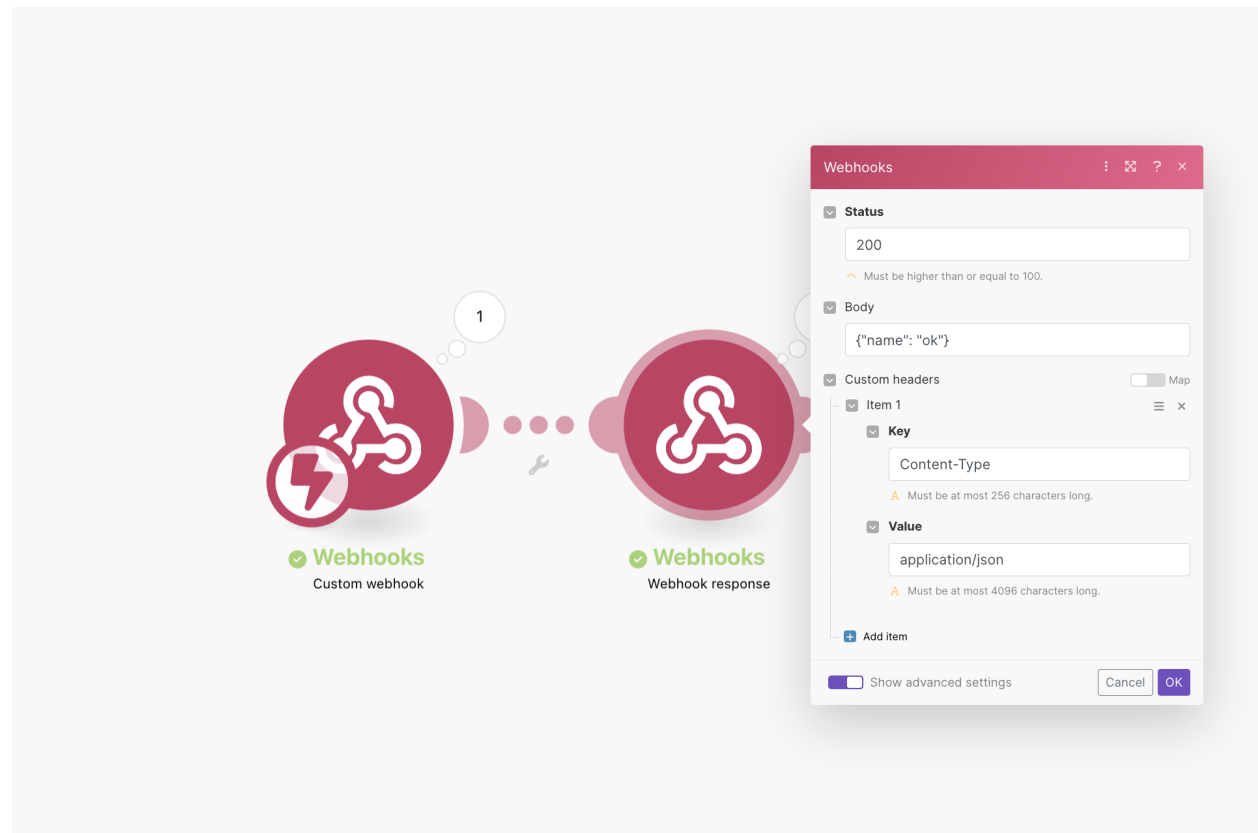
[Make.com](https://make.com) to razem z Keyboard Maestro moje dwa najważniejsze narzędzia, jakie wykorzystuję do budowania procesów, które w dużym stopniu realizowane są niezależnie ode mnie, lub wspomagają mnie w codziennych obowiązkach.

Zasada działania tej platformy jest bardzo prosta - pozwala na wizualne łączenie ze sobą usług i przesyłaniu pomiędzy nimi danych. Pomimo tego, że początkowo taka koncepcja może Ci się nie podobać, ponieważ odbiega pod wieloma względami od tego, co oferuje pisanie kodu, tak korzyści są pozytywnie przytłaczające. Największą z nich jest czas. A to w przypadku rozwijanych na Twoje potrzeby usług, jest niesamowicie ważne. Raz jeszcze podkreślam - chodzi o to aby oszczędzać Twój czas i energię a nie szukać dodatkowych miejsc w których będziesz je zużywać.

Podobnie jak w przypadku Keyboard Maestro, mój kurs Make znajdziesz na eduweb.pl. Tymczasem skupimy się na jej najważniejszych aspektach, na które potrzebujesz zwrócić szczególną uwagę.

Webhooki

Na powyższym przykładzie makra Keyboard Maestro w którym tworzyłem zapytanie http z pomocą CURL, znajduje się adres webhooka w domenie hook.eu1.make.com. Wynika to z tego, że utworzyłem go właśnie z pomocą tej platformy. Cały scenariusz wyglądał tak:

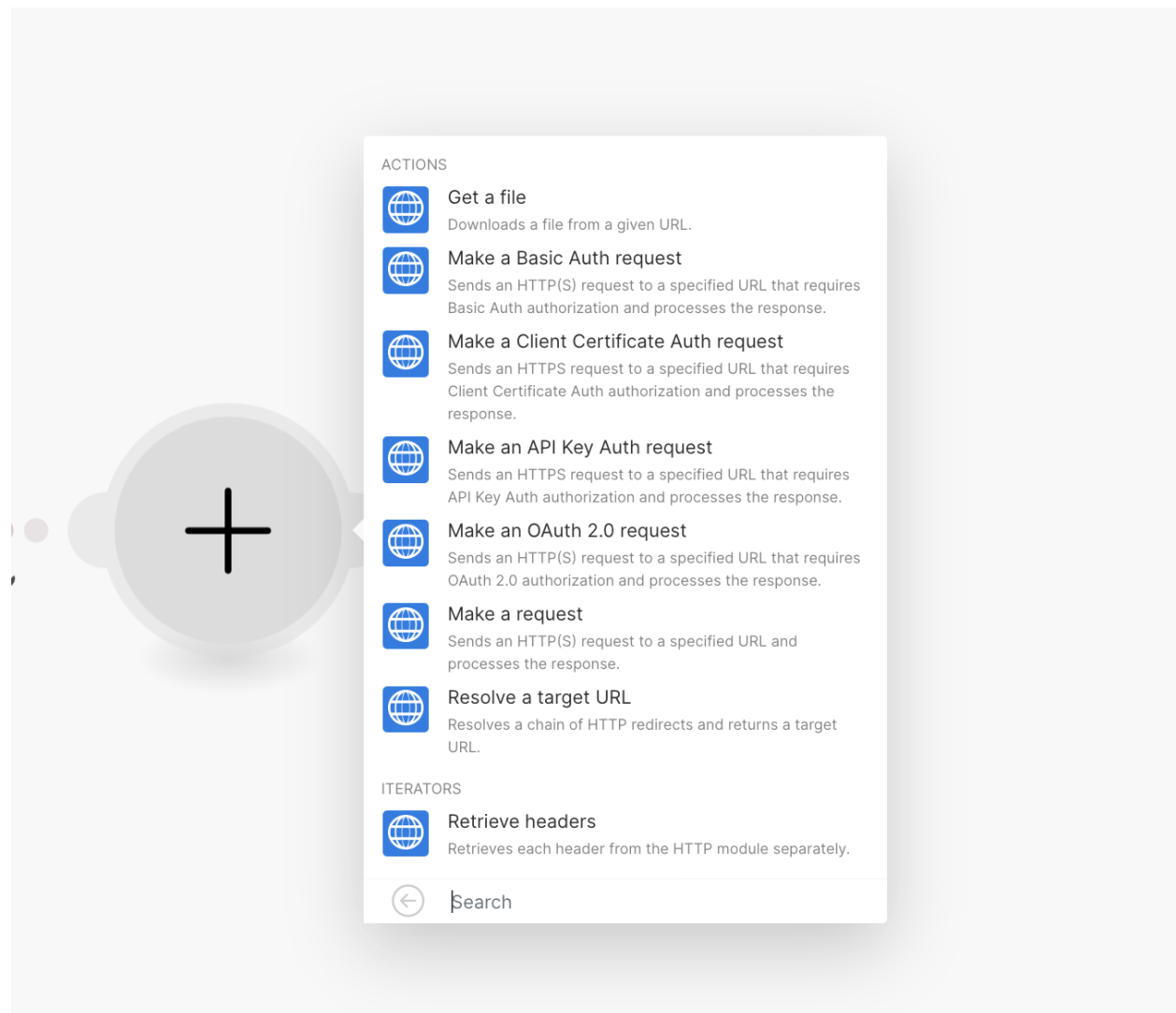


Jak widzisz jest to bardzo prosty scenariusz uwzględniający dwie akcje - odebranie danych przesłanych na webhook oraz przesłanie odpowiedzi. W dodatku mam tutaj pełną kontrolę zarówno nad statusem odpowiedzi, jego zawartością a nawet nagłówkami które zostaną zwrócone.

Co więcej, pomiędzy tymi akcjami mogę umieścić kolejne, zwyczajnie poprzez dodanie następnej "kulki". Efekt? W ciągu kilku sekund mogę wysłać maila, powiadomienie na Slacka, przesłać plik, opublikować wpisy w social media czy nawet wysłać newsletter! To wszystko bez napisania praktycznie jakiegokolwiek linii kodu.

Zapytania HTTP

Idąc teraz dalej, Make posiada jeszcze jeden moduł - HTTP, który zawiera akcje umożliwiające wysyłanie zapytań HTTP i tym samym ... połączenie się z dowolnym API! Także niezależnie od tego z jaką usługą potrzebujesz się skontaktować, nie musi ona posiadać natywnej integracji z platformą Make. Wystarczy że udostępnia API do którego możesz się podłączyć:

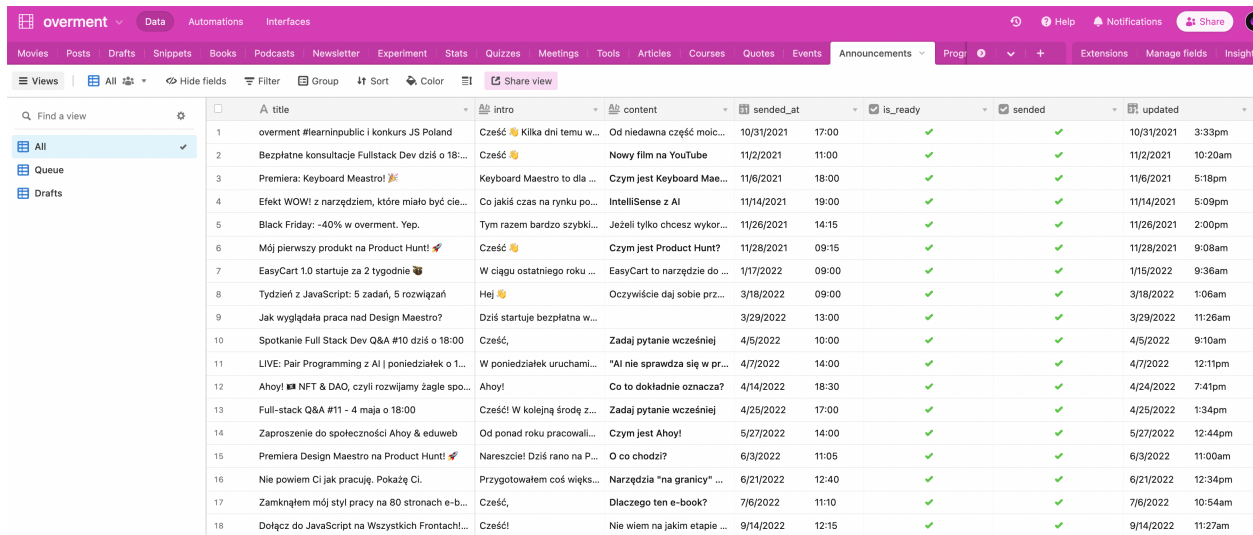


Jak widzisz mamy tutaj kilka akcji do obsługi różnego rodzaju połączeń. Ponownie mamy niemal pełną kontrolę nad sposobem komunikacji oraz formatami przesyłania danych.

Oznacza to, że w kilka minut, możesz stworzyć scenariusz, który będzie przesyłany pomiędzy szeregiem usług. Jedną z najważniejszych w kontekście tworzenia takich scenariuszy jest Airtable. To rodzaj bazy danych, który udostępnia automatycznie

wygenerowane API. Co więcej to API może komunikować się z Twoją aplikacją Node.js albo Next.js oraz scenariuszami Make. Inaczej mówiąc, Airtable może stać się miejscem wymiany informacji pomiędzy Twoją aplikacją a zewnętrznymi usługami i nawet makrami Keyboard Maestro.

Przykładowo tutaj znajduje się baza wysyłek mojego newslettera. W razie potrzeby mogę wrócić szybko do dowolnej treści, którą wysyłałem na przestrzeni ostatnich lat.



	title	intro	content	sent_at	is_ready	sended	updated
1	overment #learningpublic i konkurs JS Poland	Cześć 🙌	Od niedawna część moich...	10/31/2021 17:00	✓	✓	10/31/2021 3:33pm
2	Bezpłatne konsultacje Fullstack Dev dziś o 18:00	Cześć 🙌	Nowy film na YouTube	11/2/2021 11:00	✓	✓	11/2/2021 10:20am
3	Premiera: Keyboard Maestro! 🎉	Keyboard Maestro to dla ...	Czym jest Keyboard Mae...	11/6/2021 18:00	✓	✓	11/6/2021 5:18pm
4	Efekt WOW! z narzędziem, które miało być cie...	Co jakiś czas na rynku po...	IntelliSense z AI	11/14/2021 19:00	✓	✓	11/14/2021 5:09pm
5	Black Friday: -40% w overment. Yep.	Tym razem bardzo szybki...	Jeżeli tylko chcesz wykor...	11/26/2021 14:15	✓	✓	11/26/2021 2:00pm
6	Mój pierwszy produkt na Product Hunt! 🎉	Cześć 🙌	Czym jest Product Hunt?	11/28/2021 09:15	✓	✓	11/28/2021 9:08am
7	EasyCart 1.0 startuje za 2 tygodnie 🎉	W ciągu ostatniego roku ...	EasyCart to narzędzie do ...	1/17/2022 09:00	✓	✓	1/15/2022 9:36am
8	Tydzień z JavaScript: 5 zadań, 5 rozwiązań	Hej 🙌	Oczywiście daj sobie prz...	3/18/2022 09:00	✓	✓	3/18/2022 1:06am
9	Jak wyglądała praca nad Design Maestro?	Dzisiaj startuje bezpłatna w...		3/29/2022 13:00	✓	✓	3/29/2022 11:26am
10	Spotkanie Full Stack Dev Q&A #10 dziś o 18:00	Cześć,	Zadaj pytanie wcześniej	4/5/2022 10:00	✓	✓	4/5/2022 9:10am
11	LIVE: Pair Programming z AI poniedziałek o 1...	W poniedziałek uruchami...	"AI nie sprawdza się w pr...	4/7/2022 14:00	✓	✓	4/7/2022 12:11pm
12	Ahoy! 🎉 NFT & DAO, czyli rozwijamy zagle spo...	Ahoy!	Co to dokładnie oznacza?	4/14/2022 18:30	✓	✓	4/24/2022 7:41pm
13	Full-stack Q&A #11 - 4 maja o 18:00	Cześć! W kolejną środę z...	Zadaj pytanie wcześniej	4/25/2022 17:00	✓	✓	4/25/2022 1:34pm
14	Zaproszenie do społeczności Ahoy & eduweb	Od ponad roku pracowali...	Czym jest Ahoy!	5/27/2022 14:00	✓	✓	5/27/2022 12:44pm
15	Premiera Design Maestro na Product Hunt! 🎉	Nareszcie! Dziś rano na P...	O co chodzi?	6/3/2022 11:05	✓	✓	6/3/2022 11:00am
16	Nie powiem Ci jak pracuję. Pokażę Ci.	Przygotowałem coś więk...	Narzędzia "na granicy" ...	6/21/2022 12:40	✓	✓	6/21/2022 12:34pm
17	Zamknąłem mój styl pracy na 80 stronach e-b...	Cześć,	Dlaczego ten e-book?	7/6/2022 11:10	✓	✓	7/6/2022 10:54am
18	Dołącz do JavaScript na Wszystkich Frontach!	Cześć!	Nie wiem na jakim etapie ...	9/14/2022 12:15	✓	✓	9/14/2022 11:27am

Wiesz jednak co jest w tym najbardziej ciekawe? Że ten newsletter wysyłany jest w sposób, który nie wymaga ode mnie logowania do systemu Newslettera (Sendgrid). Flow wygląda tak:

- z pomocą opcji text expander Keyboard Maestro, generuję sobie "szablon wiadomości". Jest to po prostu kilka sekcji, które muszę uzupełnić

```
Scheduled at:
2022/09/25 23:58
TITLE----

INTRO----

CONTENT----

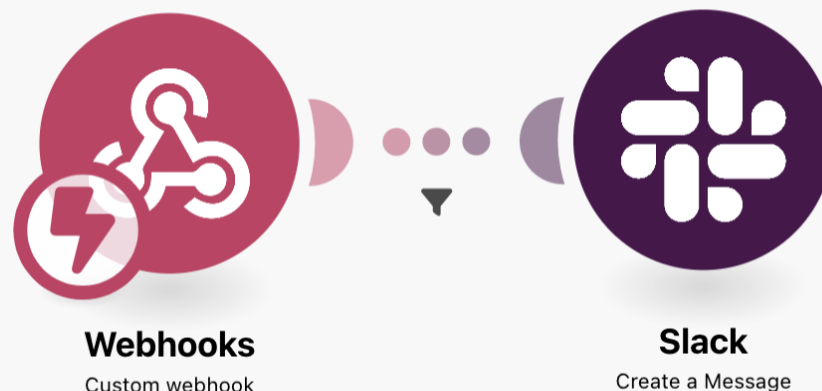
-----
#newsletter #announcement
```

- następnie po napisaniu treści, wciskam skrót klawiszowy który zabiera całą zawartość z iA Writer i przesyła ją do Airtable oraz przygotowuje szablon HTML na podstawie składni markdown. Ja otrzymuję powiadomienie, ze screenem newslettera do zatwierdzenia.
- gdy je zatwierdzę, kampania zostaje potwierdzona i wysłana przez Sendgrid z odpowiednimi ustawieniami.

Tylko ta jedna aktywność oszczędza mi przynajmniej kilka godzin w ciągu miesiąca.

Taki rodzaj połączenia sprawia, że fakt że programujesz, nie tylko pozwala Ci rozwijać aplikacje, lecz zaczynasz bardzo efektywnie **programować z wykorzystaniem wielu aplikacji i usług!** Nagle okazuje się że możesz nawiązać połączenie z systemami mailowymi, usługami konwertującymi dane, tworzącymi tłumaczenia (np. HappyScribe) czy nawet bazami danych!

Chciałbym podkreślić tutaj coś jeszcze. Fakt że Make udostępnia zarówno Webhooki jak i moduł HTTP umożliwiający przesłanie danych na dowolny adres, oznacza że **scenariusze Make mogą komunikować się ze sobą**. Możesz więc tworzyć automatyzacje pośrednie, które będą realizować jakąś powtarzalną logikę. W moim przypadku są to automaty odpowiedzialne za przesyłanie wiadomości na Slacka w ustalonym wcześniej formacie:



Super prosty scenariusz - przesłane dane, wysłane powiadomienie. Efekt? Nie muszę za każdym razem konfigurować np. avatara i nazwy bota który wysyła wiadomość. Dzieje się to w jednym miejscu, bo wszystkie pozostałe scenariusze kontaktują się właśnie z tym.

Jednocześnie jest w tym uwzględniona kolejna wskazówka, mówiąca o tym, że tworzone przeze mnie makra i scenariusze zwykle są bardzo proste. Ułatwia mi to ich tworzenie oraz późniejsze utrzymywanie. Bo tak, to również musisz brać pod uwagę. Takie scenariusze lub tworzone przez Ciebie aplikacje mogą przestać działać lub zawierać błędy. Oznacza to że na pewnym etapie spadnie na Ciebie ich naprawa. Dobrze więc zadbać o to, aby takie sytuacje miały miejsce jak najrzadziej. **Moja strategia polega tutaj na upraszczaniu i unikaniu złożoności.**

Własne moduły Make!

Podobnie jak Raycast, Make udostępnia świetny interfejs, z pomocą którego utworzysz swoje własne moduły. W niektórych sytuacjach możesz nawet przejść etap weryfikacji i opublikować taki moduł. W moim przypadku korzystam jednak z opcji modułów prywatnych (do których i tak mogę zaprosić inne osoby).

Make Apps

Search...

About the apps builder

How to read the documentation

Your first app >

VS Code extension >

Debugging your app >

Integromat DevTool >

Best practices >

Useful Resources

App Visibility

App Review >

Changes and versioning >

APP STRUCTURE

Base >

Connections >

Webhooks >

Modules >

Action >

Search

Triggers (polling)

Modules

Copy link

There are six basic types of modules:

Action	Use if the API endpoint returns a single response. Examples are Create a book, Delete a book or Get a Book.
Search	Use if the API endpoint returns multiple items. An example is List Books that will find specific books according to search criteria.
Trigger (polling)	Use if you wish to watch for any changes in your app/service. Examples are Watch a New Book, which will be triggered whenever a new book has been added to the library.
Instant Trigger (webhook)	Use if the API endpoint has a webhook available (dedicated or shared). Example is Watch a New Event.
Universal	Use if you want to enable users to perform an arbitrary API call to the service. Examples are Make an API Call and Execute a GraphQL Query.
Responder	Use if you need to send a processed data back to a webhook.

←

Previous
Not attached

Next
Action

→

Last modified 4d ago

<https://docs.integromat.com/apps/app-structure/modules>

Przygotowanie takiego modułu jest stosunkowo proste. Składa się z:

- podstawowych ustawień połączenia (główny endpoint, nagłówki itd.)
- połączenia (wymagane np. w przypadku gdy posiadasz konta użytkowników)
- webhooki (akcje które reagują natychmiast i odbierają dane z Twojej aplikacji)
- moduły (akcje które pozwalają wchodzić w interakcję z danym API, np. Twojej aplikacji)

Są to najważniejsze elementy modułu Make. Podobnie jak w Design Maestro, tworzenie modułów sprowadza się do opisywania obiektami JSON ich poszczególnych zachowań. Wszelkie informacje znajdziesz w dokumentacji. Polecam Ci również zajrzeć do zakładki `my-apps`. Znajdziesz tam kilka przykładów aplikacji, które możesz wykorzystać jako inspirację do stworzenia własnych.

Przykładem stworzonego przeze mnie modułu jest EasyBanner. To opracowana przeze mnie usługa odpowiadająca za przygotowanie bannerów i grafik. Jej główna część to aplikacja node.js połączona z kilkoma dodatkowymi narzędziami. Jednak sama aplikacja udostępnia API, które wykorzystałem do przygotowania modułu Make.



Create user Action

Create user and generate API key



Extract Colors Action

Extract colors from images. Supports GIF/JPG/PNG/SVG



Render PDF Action

Render a new pdf from HTML / CSS



Render PNG / JPG Action

Render a new png from HTML / CSS



URL Screenshot Action

Take a screenshot of a given url



Create a new Module

Efekt? Makro Keyboard Maestro pytając mnie o kilka detali, wysyła zapytanie do Make, gdzie przygotowywana jest dla mnie grafika. Gotowy plik png trafia na mój komputer i zostaje zapisany w schowku (w zależności od ustawień albo jako adres URL albo jako graficzna zawartość schowka). Magia!

Moduły Make możesz wykorzystać nie tylko w połączeniu ze swoimi aplikacjami. Nic nie stoi na przeszkodzie aby połączyć je z Webhookami z Twoich scenariuszy Make i w ten sposób ułatwiać sobie wchodzenie z nimi w interakcję. To samo możesz zrobić z API innych usług z których korzystasz, budując "nieoficjalne" integracje z platformą Make.

Twój własny asystent

Niewykluczone że na tym etapie zastanawiasz się, co możesz zrobić ze wszystkimi wskazówkami i przykładami, które do tej pory opisałem. W związku z tym, że mówimy tutaj o całkiem dużej ilości zarówno informacji jak i pracy, którą potencjalnie masz do wykonania. To co widzisz tutaj, to mały fragment tego, co budowałem przez ostatnie

dwa lata. Trudno byłoby pokazać wszystko, ale myślę że udało mi się uwzględnić najważniejsze obszary, dzięki czemu **znacząco skróciłem drogę**, którą masz przed sobą.

Nie zmienia to jednak faktu, że prawdopodobnie najlepszą możliwą strategią jaką możesz podjąć, jest stopniowe zapoznawanie się z opisanymi przeze mnie narzędziami a następnie podejmowanie małych kroków do tego, aby tworzyć własne skróty, makra i automatyzacje. Usprawniając każdego tygodnia jakiś mały proces, czymkolwiek by to nie było, **zaczyniesz zyskiwać coraz więcej przestrzeni do tego, aby zdobywać jej jeszcze więcej.**

Prawdopodobnie najlepszym z możliwych sposobów jest zarezerwowanie sobie bloku czasu w każdym tygodniu na to, aby wdrażać kolejne fragmenty wiedzy z powyższego e-booka i rozwijać Twoją znajomość narzędzi takich jak Raycast, Make czy Keyboard Maestro.

Myślę że całkiem interesującym kierunkiem, jest stworzenie swojego własnego asystenta. W ten sposób, będziesz w stanie łatwiej określić zarówno obszary jak i procesy, które mogą zostać oddane w jego ręce. Po prostu - wyobraź sobie, że zatrudniasz pracownika na kilka godzin w miesiącu i płacisz mu 300-500 zł. Jednocześnie jest to Twój budżet na narzędzia i usługi, które zaczną oszczędzać Ci czas. Oczywiście nie musisz wydawać wszystkiego od razu. Chodzi wyłącznie o to aby przestawić Twoje myślenie na konkretne tory i podsuwać Ci bardzo konkretne wskazówki.

Skąd to wiem? Bo sam przechodziłem przez to kilka razy i aktualnie współpracuję z kilkoma takimi asystentami. Poznaj mój robotyczny zespół:



Ellie



Luke



Claire



Nicky



Jenny



Rose



Tony

Aby ułatwić sobie tworzenie automatyzacji, zamiast nadawać kategorie, przyporządkowałem je do stworzonych przeze mnie Avatarów, którym nadałem imiona. Każdy z nich obecny jest w Keyboard Maestro, Raycast, Make, Shortcuts czy nawet w Make.

Każdy z nich posiada swój adres e-mail (mailhook) i każdy z nich ma określoną listę zadań do realizacji, która stopniowo się poszerza. Co więcej, nawet Tony dołączył do nas kilka miesięcy temu!

Teraz nie muszę się już zastanawiać jak kategoryzować i porządkować moje makra czy scenariusze. Po prostu odpowiadam sobie na pytanie, kto z nich jest odpowiedzialny za wykonanie danego zadania.

Ciekawostka: Każdy z tych robotów może komunikować się ze sobą. Np. Rose może poprosić Nicky o przygotowanie grafiki do newslettera i może się to odbyć bez mojego udziału!

Pomysły na automatyzacje

Mógłbym w tym momencie opisać Ci wszystkie moje aktualne automatyzacje oraz pomysły, których jeszcze nie wdrożyłem. Bez wątpienia fundamentalnym elementem całej gry o której tutaj mówimy, są właściwe pomysły. Z łatwością przygotujesz skomplikowane makra, z których nie będziesz korzystać. Sztuka polega na tym aby skupić się wyłącznie na tych, które faktycznie będą oszczędzać Ci czas. Naturalnie przychodzi to z praktyką ale ... pamiętasz co mówiłem o Twoim otoczeniu i jego roli?

Dlatego nie zasypię Cię teraz moimi pomysłami, ponieważ myślę że i tak masz ich już całkiem sporo. Zamiast dać Ci rybę, dam Ci wędkę i podzielę się z Tobą moimi źródłami wiedzy i inspiracji z obszaru automatyzacji.

Dodam tylko, że to co tutaj widzisz, to dość unikatowe podejście, którego nie spotkałem nigdy wcześniej. Jest to rezultat moich własnych przemyśleń oraz doświadczeń, połączony z tym, co spotkałem w niżej wymienionych miejscach.

- forum.keyboardmaestro.com: Tutaj znajdziesz prawdopodobnie wszystko czego szukasz na temat Keyboard Maestro. To absolutna skarbnica wiedzy, przepełniona ludźmi którzy są gotowi Ci pomóc. To taki stackoverflow dla KM.
- talk.automators.fm: kolejne miejsce w którym znajdziesz mnóstwo osób zainteresowanych automatyzacją w systemie macOS. Regularnie tam zaglądam

- <https://www.makerpad.co>: genialne community skupione wokół narzędzi no-code i low-code. Warto zajrzeć do sekcji tutoriali
- <https://www.raycast.com/community> genialnie rozwijany kanał na Slacku. W temacie Raycast znajdziesz tutaj wszystko czego potrzebujesz. Połącz to z dokumentacją i oficjalnym repozytorium Raycasta i zacznij tworzyć.
- <https://brettterpstra.com> ten blog to absolutne złoto pod kątem również niszowych aplikacji i technik
- <https://club.macstories.net> tak jakościowych wpisów na temat organizacji pracy w systemie MacOS nie spotkałem nigdzie indziej
- <https://ahoy.so/> no i oczywiście Ahoy, czyli miejsce które mam przyjemność współtworzyć.

Dodaj sobie te wszystkie strony nie tylko do zakładek ale najlepiej w miejsce, które sprawi że z pewnością będziesz tam regularnie wracał. To absolutnie najlepsze źródła na jakie udało mi się do tej pory trafić. Każdego dnia czerpię z nich mnóstwo inspiracji i poznałem tam wielu wspaniałych ludzi. Z niektórymi z nich utrzymuję stały kontakt i Tobie polecam to samo.

To jest właśnie praktyczne kształtowanie swojego otoczenia i tego, z kim spędzasz najwięcej czasu. W moim przypadku są to ludzie, którym zależy na tym, aby rozwijać swój styl pracy, który sprawia że codzienne obowiązki stają się przyjemne. Bo tak, praca może być przyjemna, zabawna i sprawiać mnóstwo radości a w tym wszystkim - być połączona z tym, co dla Ciebie ważne.

Kilka inspiracji dla Ciebie

Powiedziałem Ci, że nie zasypię Cię pomysłami oraz przykładami zaawansowanych automatyzacji. Nie zmienia to jednak faktu, że dobrze byłoby nie zostawiać Cię z niczym, nawet pomimo tego, że już teraz masz o czym myśleć.

Przeprowadzę Cię teraz przez kilka przykładów makr i automatyzacji, które możesz praktycznie natychmiast wdrożyć, lub przynajmniej rozpocząć pracę nad nimi. Oto ich lista:

- modyfikatory i transformatory tekstu
- zestaw skrótów klawiszowych ułatwiających Ci rozwój aplikacji

- modyfikatory nazw plików i katalogów
- makra uruchamiające to, czego potrzebujesz do pracy
- makra do łatwiejszego trackowania czasu
- automatyzacje raportujące Twój czas pracy
- automatyzacje i reguły w Twojej skrzynce pocztowej
- listy snippetów kodu, po które często sięgasz
- snippety text expandera ułatwiające pisanie
- automatyczne powiadomienia na Slacku o ważnych dla Ciebie zdarzeniach
- makra ułatwiające Ci zarządzanie repozytorium
- automatyzacje obserwujące Twoje ulubione blogi
- automatyzacje obserwujące status Twoich aplikacji
- organizacja Twoich źródeł wiedzy i artykułów
- jeżeli tworzysz bloga - zautomatyzuj to, co nie jest związane bezpośrednio z tworzeniem. A sam proces tworzenia spróbuj sobie ułatwić
- stwórz system ułatwiający Ci zarządzanie domowym budżetem (sam korzystam z Revoluta)
- zapoznaj się z możliwościami aplikacji Shortcuts kilka użytecznych dla siebie
- sprawdź jakie integracje i API posiadają aplikacje z których korzystasz i zastanów się, co możesz z tym zrobić
- zwróć szczególną uwagę na to, aby łączyć ze sobą różne usługi i aplikacje. (Może Github może jakoś połączyć się z Twoją listą zadań?)
- zapoznaj się z Airtable. Czy masz jakieś informacje które chcesz przechowywać w takiej bazie?
- pracujesz na kilku monitorach? Zaprojektuj makra które ułatwią Ci pracę na nich
- stwórz własnego bota na Discordzie / Telegramie czy Slacku, który będzie informował Cię o tym, co dla Ciebie istotne
- zastanów się z jakich usług i serwisów korzystasz oraz czy możesz w jakiś sposób to sobie ułatwić lub zautomatyzować

- znasz Notion? Sprawdź API tej aplikacji i zastanów się jak możesz je wykorzystać. Ja stworzyłem swój własny Dashboard, który gromadzi mi istotne dla mnie informacje (np. listy spotkań, aktywności, ważne linki, wiadomości i statystyki)
- wykonujesz cyklicznie jakieś aktywności, które wymagają od Ciebie podjęcia powtarzalnego działania? Może mogą zrobić to za Ciebie roboty?
- utwórz Avatary swojego robota. Pobaw się tym!

Lista tych ogólnych wskazówek, powinna dodatkowo nakierować Twoją uwagę na obszary, które w połączeniu z całą dotychczasową wiedzą zawartą w tym e-booku, zaprowadzi Cię w bardzo ciekawe miejsca. W ten sposób masz możliwość doświadczyć tego, co można nazwać już moją codziennością. Chciałbym tylko podkreślić, że w tym wszystkim nie chodzi ani o makra ani automatyzacje. Chodzi o to aby ułatwić Ci codzienne funkcjonowanie. Jeżeli powyższe pomysły nie wniosą Ci wartości, pomiń je. Spróbuj wymyślić własne - takie, które będą dostosowane do Ciebie i Twoich celów.

Twoja Armia Robotów

Na sam koniec chciałbym podzielić się z Tobą czymś jeszcze, w czym według wielu jestem całkiem dobry - inspiracją.

Every great software developer, for example, now has an army of robots working for him at nighttime while he or she sleeps, after they've written the code, and it's cranking away. ~ Naval Ravikant

Według Navala Ravikanta każdy świetny developer posiada już swoją własną armię robotów, która pracuje dla niego dzień i noc. To o czym tutaj mówi, nie jest próbą przewidywania przyszłości - to fakt. Już teraz możesz budować "roboty" które będą pracować dla Ciebie cały czas oraz pomagać Ci w codziennych aktywnościach. W moim przypadku mówimy już o setkach godzin pracy, które oszczędzam dzięki odpowiedniej organizacji mojego otoczenia oraz **sprytnemu wykorzystaniu technologii**.

Fakt, że potrafisz programować to w obecnych czasach absolutna supermoc. Jednak co ciekawe, w zupełności nie chodzi tutaj o to, aby "wymiać" w wybranych technologiach. W genialny sposób oddaje to cytaty z książki "Getting Real"

The best designers and the best programmers aren't the ones with the best skills, or the nimblest fingers, or the ones who can rock and roll with Photoshop or their environment of choice, they are the ones that can determine what just doesn't matter. That's where the real gains are made.

Jak widzisz, w byciu najlepszym w swojej dziedzinie w zupełności nie chodzi o to aby najlepiej posługiwać się narzędziami czy językami programowania. Chodzi o to aby trafnie określać **co w danej chwili nie ma znaczenia**. W kontekście rozwoju zawodowego dla mnie, jest to **najważniejsza umiejętność jaką rozwijam**.

Ponownie wrócimy tutaj do Navala Ravikanta, który powiedział jeszcze:

Forget 10x programmers. 1,000x programmers really exist, we just don't fully acknowledge it. See @ID_AA_Carmack, @notch, Satoshi Nakamoto, etc.

i dalej:

For example, a good software engineer, just by writing the right little piece of code and creating the right little application, can literally create half a billion dollars' worth of value for a company. But ten engineers working ten times as hard, just because they choose the wrong model, the wrong product, wrote it the wrong way, or put in the wrong viral loop, have basically wasted their time. Inputs don't match outputs, especially for leveraged workers.

Wszystkie powyższe cytaty mówią tylko i wyłącznie o jednym: **granica możliwości leży dalej niż ktokolwiek jest w stanie sobie to wyobrazić a sposób dotarcia do niej polega na czymś innym, niż powszechnie się uważa**.

Pomyśl o tym,
Adam

