LogInt – dokumentacja wdrożeniowa

# Instrukcja instalacji:

Produkt LogInt wdrożony jest jako integracja środowiska Docker wraz z lokalnym emulatorem urządzenia mobilnego. Należy zatem zainstalować obydwa środowiska w celu poprawnego funkcjonowania systemu LogInt.

1. Środowisko Docker:
   1. Zaintaluj Docker Engine
   2. Zainstaluj Docker Compose
2. Instalacja Emulatora:

Zainstaluj środowisko andorid studio tak jak przedstawiono w rozdziale „Przygotowanie środowiska wykonywalnego programu LogInt dla systemu Linux”

1. Instalacja systemu LogInt:
   1. Sklonuj repozyturium LogInt
      1. git clone <https://github.com/micwuj/LogIntProject.git>

# Przygotowanie środowiska wykonywalnego programu LogInt dla systemu Linux

Instalacja Android Studio wraz z narzędziami:

1. korzystając z terminala otwórz folder: LogIntProjext/ConfigurationScripts
2. *chmod +x AndroidStudioSetup.sh*
3. *./AndrodiStudioSetup.sh*

*Android Studio można róznież zainstalować za pomocą Snap Store*

Program AndroidStudio powinien zostać zainstalowny i uruchomiony, wykonaj w nim następujące kroki:

1. Utwórz nowy emulator
   1. More actions > Virtual device manager > Create new virtual device > Pixel 8 Pro
   2. Jako system image wybierz UpsideDownCake, API 34
   3. Avd name ustaw na executoner

Nowy emulator powienien pokazać się na liście

1. Uruchom skrypt AvdSetup.sh
2. Zainstaluj Android SDK Command-line tools
   1. More Actions > SDK Manager > SDK Tools > Android SDK Command-line Tools
3. W programie Android Studio:
   1. More Actions > Virtual device manager
   2. Uruchom emulator o nazwie executoner
   3. W ustawieniach urządzenia włącz opcje programisty:
      1. Włącz lokalizację kursora
      2. Włącz pokazywanie kliknięć
      3. Włącz 3-guzikowy sytem nawigacji
      4. Włącz USB Debugging
      5. Włącz Wireless Debugging

Android Debug Bridge – jest to narzędzie przeznaczone do sterowania emulatorem za pomocą termianal. Jest ono również niezbędne do egzekucji skryptów wykonywujące integracje. Aby go urzywać twoje urządzenie musi wspierać wirtualizację. Możesz to sprawdzić za pomocą polecenia:

*kvm-ok*  
Jeżeli twoje urządzenie spełnia wymagania, otrzymasz następujący komunikat:  
*INFO: /dev/kvm exists*  
*KVM acceleration can be used*

# Instrukcja Uruchomienia

Środowisko LogInt docelowo działa całodobowo, oraz nieustannie egzekuje integracje przy otrzymaniu nowych raportów z bazy danych. W celu uruchomienia systemu LogInt należy wykonać poniższe kroki:

1. Uruchomienie środowiska kreacji:
   1. Przenieś się do katalogu roboczego systemy LogInt
      1. cd LogIntProject
   2. uruchom docker compose
      1. docker-compose up –build
2. Uruchomienie środowiska egzekucji:
   1. Uruchom program Android Studio
      1. More actions > Virtual device manager
         1. Uruchom wcześniej przygotowany emulator o nazwie „executoner”

Po uruchomieniu obu środowisk emualtor dostępny jest w wersji okienkowej na urządzeniu użytkownika, a obszar kreacji dostępny jest pod adresem: <http://localhost:8000>

Aby jednakowo korzystać ze środowiska kreacji oraz środowiska wykonywalnego rekomendowane jest używanie dwóch emulatorów (jeden do wykonywania egzekucji, a drugi jako pomoc przy tworzeniu kroków w skrypcie wykonywalnym). Drugi emulator zaleca się stworzyć poprzez kopię pierwszego (executoner), oraz zmienienie jego nazwy na odpowiadającą użytkownikowi. Zaleca się również zmniejszenie ilości ramu przypisanego do niego, w celu optymalizacji działania całego środowiska LogInt.

# Podręcznik użytkowania:

## Dodanie nowego źródła:

1. Przejdź do zakładki Sources
2. New source
3. Podaj nazwę źródła oraz link do api bazodanowego

## Dodanie nowej integracji:

1. Przejdź do zakładki home
2. Add Integration
3. Uzupełnij pola w oknie pop up
4. Save

## Edycja istniejącej integracji:

Dotychczas utworzone integracje możesz eytować za pomocą przycisku more info. Możliwe akcje to:

1. Deactivate – tymczasowo deaktywuje integracje
2. Edit (niebieski przycisk) - pozwala zmienić nazwę, skrypt, czy źródło danych
3. Delete (czerwony przycisk) - usuwa integrację - brak możliwośći przywrócenia
4. Add Driver
   1. Funkcja Add Driver jest niezwykle użyteczna gdy dla danej integracji przypisać będziemy mogli kilku kierowców. Wtedy wszystkie akcje podczas egzekucji będą wykonywane tak samo, ale ze zmienionymi danymi logowania do aplikacji. Funkcja ta pozwala uniknąć redundancji integracji w zakładce Home

## Podgląd historii:

W zakładce „history” znajduje się zapis rejestrów jakich działań dokonano w środowisku kreacji (utworzenie integracji / dodanie źródła / dodanie kierowcy). Jest to funkcjonalność szczególnie przydatna gdy systemem będzie zarządzał więcej niż jeden administrator.

## Podgląd raportów:

Raporty są to zapisy rejestrów pochodzące z egzekucji integracji. Pozwalają one weryfikować jakie dane oraz z jakiej przyczyny nie zostały przekazane do aplikacji kontrahenta. Zawierają one również dane z pliku json, który został pobrany ze źródła jako źródło danych do przekazania podczas egzekucji. Każdy z raportów można pobrać w formacie pdf klikając ikonę po prawej stronie w każdym wierszu.

## Dodanie nowego skryptu egzekucyjnego:

W celu poprawnej egzekucji integracji konieczny jest wcześniej przygotowany i sprawdzony skrypt. To właśnie on zostanie wywołany gdy będzie trzeba zrzutować dane z pliku json do aplikacji kontrahenta. Jego nazwa musi być taka sama jak ta podana przy tworzeniu integracji. W obecnej wersji skrypty pisane są ręcznie przez administratora. Zawierają one głównie polecenia pakietu Android Debug Bridge. Poprawny i przetestowany skrypt należy umieścić w folderze: LogIntProject > Executoner > scripts

## Dodanie nowej aplikacji (pliku apk):

Analogicznie jak w przypadku skryptu egzekucyjnego aplikacja będzie używana podczas każdej egzekucji i jest ściśle powiązana ze skryptem. Dodając nową wersję aplikacji najpewniej konieczne będzie również stworzenie lub modyfikacja skryptu. Plik źródłowy z rozszerzeniem .apk należy umieszczać w folderze: LogIntProject > Executoner > apk\_files

# Diagram wdrożenia:

