

LogInt

Dokument ten ma na celu ułatwić odbiorcy zapoznanie się z terminologią projektową oraz niuansami technologicznymi, które w tym projekcie się pojawiają. Ma on charakter wprowadzenia do całej logiki systemowej, oraz przedstawienie jak poszczególne komponenty są ze sobą nierozzerwalnie powiązane.

Spis treści

Terminologia.....	3
Twierdzenia ogólne.....	3
Główne funkcjonalności.....	3
Wykaz komponentów.....	3
Integracja z systemami klienta.....	3
Diagramy.....	4
Diagram klas.....	4
Diagram stanów.....	4
Diagram przypadków użycia.....	5
Diagram aktywności.....	5
Diagram wdrożenia.....	6
Diagramy plakatuowe.....	6
Instrukcja instalacji.....	7
Przygotowanie środowiska wykonywalnego programu LogInt dla systemu Linux.....	7
Instrukcja Uruchomienia.....	8
Podręcznik użytkownika.....	8
Dodanie nowego źródła:.....	8
Dodanie nowej integracji:.....	8
Edycja istniejącej integracji:.....	8
Podgląd historii:.....	8
Podgląd raportów:.....	9
Dodanie nowego skryptu egzekucyjnego:.....	9
Dodanie nowej aplikacji (pliku apk):.....	9
Dokumentacja Testowa.....	9
Moduł History:.....	9
Testy jednostkowe:.....	9
Testy integracyjne:.....	9
Moduł Home:.....	10
Testy jednostkowe:.....	10
Testy integracyjne:.....	10
Moduł Source:.....	11
Testy jednostkowe:.....	11
Testy integracyjne:.....	12
Moduł Reports:.....	12
Pokrycie kodu przez testy.....	14

Terminologia

- Integracja – przygotowana w systemie LogInt logika zawierająca: źródło danych, zakres danych (zapytanie sql), aplikację oraz skrypt egzekucyjny, mająca na celu umożliwienie automatyzacji procesu wypełniania danych w aplikacjach mobilnych kontrahentów.
- Aplikacja – przez aplikację twórcy rozumieją aplikację mobilną (lub webową) kontrahenta. To właśnie do niej dotychczas dane były wprowadzane ręcznie, a system LogInt ma umożliwić kreowanie wzorców automatycznych działań (Integracji).
- Źródło danych (source) – jest to link do api bazodanowego klienta z którego pobierane będą dane używane przy egzekucji integracji.
- Egzekucja – pojedyncze wykonanie czynności określonych w skrypcie egzekucyjnym.
- Skrypt egzekucyjny – ręcznie napisany skrypt powłoki bash definiujący jakie czynności należy wykonać aby doszło do egzekucji integracji.
- Raport – jest to plik generowany po egzekucji integracji. Zawiera on informacje jakie dane, oraz jakie te dane miały wartości podczas konkretnej egzekucji, oraz logi z środowiska wykonywalnego, opisujące jakie czynności zaszły na urządzeniu.
- Środowisko wykonywalne – emulator urządzenia mobilnego, lub podłączony do komputera smartphone.
- Historia – jest to zapis wykonanych czynności w systemie LogInt, tj. utworzenie nowej integracji, dodanie nowego źródła danych, czy dezaktywacja istniejącej integracji.

Twierdzenia ogólne

System LogInt ma na celu umożliwienie klientowi kreowanie integracji w celu automatyzacji procesu wypełniania danych w aplikacjach mobilnych kontrahentów. Konkretnie dane będą pobierane ze zdefiniowanego w integracji źródła, oraz będą one przenoszone do aplikacji za pomocą środowiska wykonywalnego poprzez instrukcje zapisane w skrypcie egzekucyjnym. Po wykonaniu egzekucji zostanie utworzony raport stanowiący: jakie dane zostały przesłane / utracone, jakie czynności zaszły w środowisku egzekucyjnym, wartość logiczną czy egzekucja integracji zakończyła się powodzeniem.

Główne funkcjonalności

- Kreacja nowych integracji – umożliwienie użytkownikowi systemu procesu utworzenia nowej integracji: wybrania źródła danych i ich zakresu, wskazanie aplikacji kontrahenta oraz skryptu egzekucyjnego.
- Dodanie nowego źródła danych – system może pobierać dane z różnych źródeł, poprzez dodawanie nowych możliwa będzie uniwersalizacja systemu dla wielu klientów, oraz dostosowanie systemu LogInt do systemu wdrożonego u klienta
- Egzekucja integracji w środowisku wykonywalnym – dostarczenie backendowej części projektu pozwalającej naocznie zweryfikowanie oraz wdrożenie integracji utworzonych w procesie kreacji

Wykaz komponentów

- Środowisko kreacji – główna część systemu LogInt, działająca w przeglądarce. To z jej poziomu klient jest w stanie definiować jakie procedury będzie miał wykonywać system.
- Środowisko egzekucji – elementu systemu odpowiedzialne za zarządzanie przychodzącymi sygnałami odnośnie egzekucji integracji na środowisku wykonywalnym (emulatorze).
- Emulator / urządzenie mobilne – finalny punkt przepływu komunikacji. Na nim środowisko egzekucji wykona procedury zdefiniowane w środowisku kreacji.

Integracja z systemami klienta

System LogInt ma na celu umożliwienie automatycznego przepływu danych gromadzonych w bazie danych klienta do aplikacji mobilnych kontrahentów. Dotychczas dane były wypełniane ręcznie co prowadziło do wielu błędów, zaniedbań obowiązków, a w konsekwencji kar. Produkt będzie wykonywał te zadania w sposób zautomatyzowany.

Diagramy

Diagram klas

Diagram klas

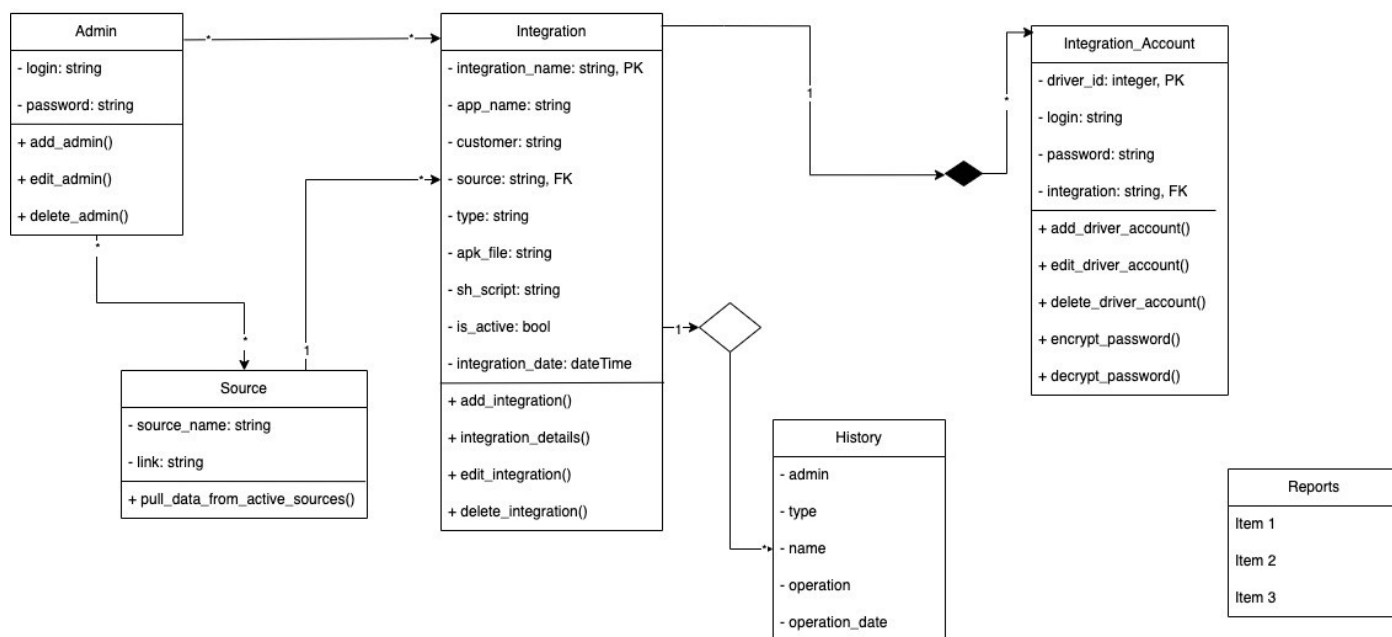


Diagram stanów

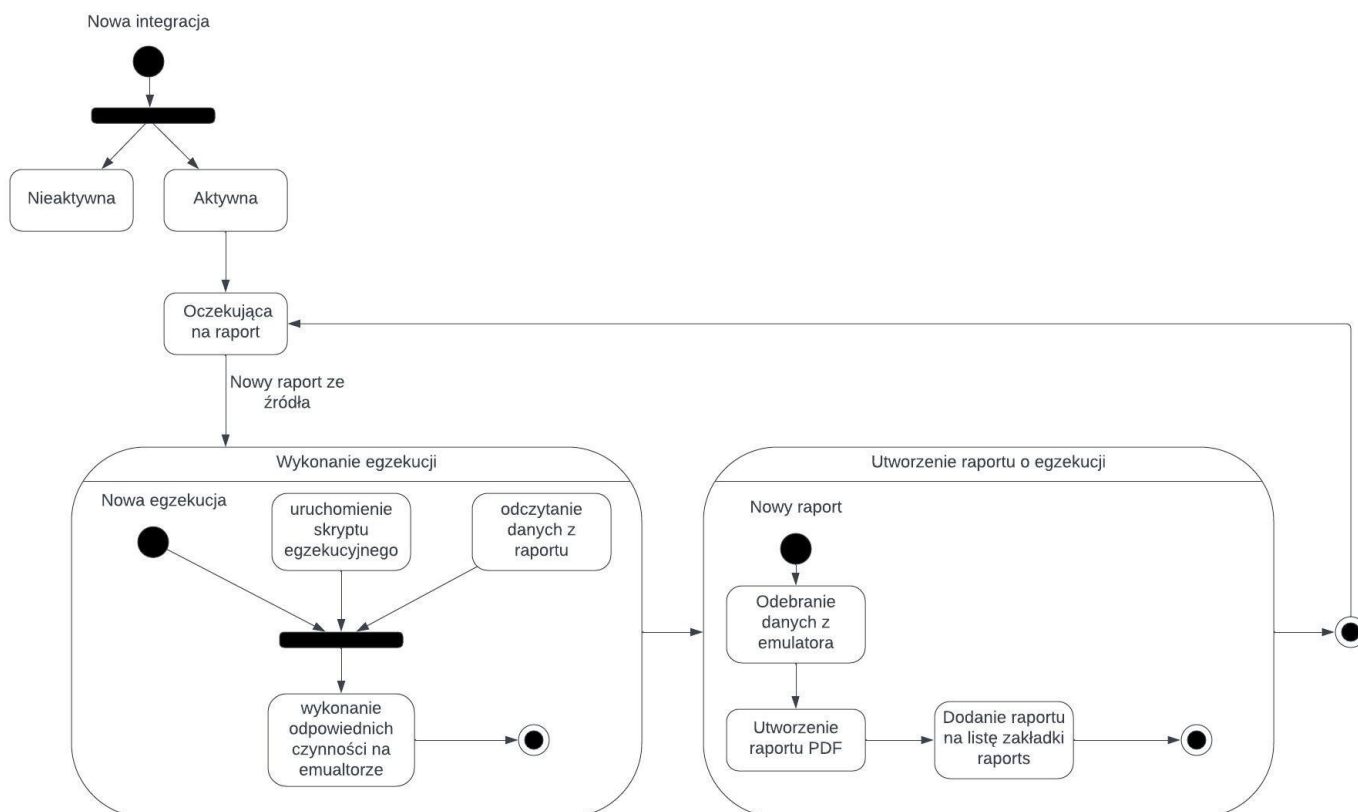


Diagram przypadków użycia

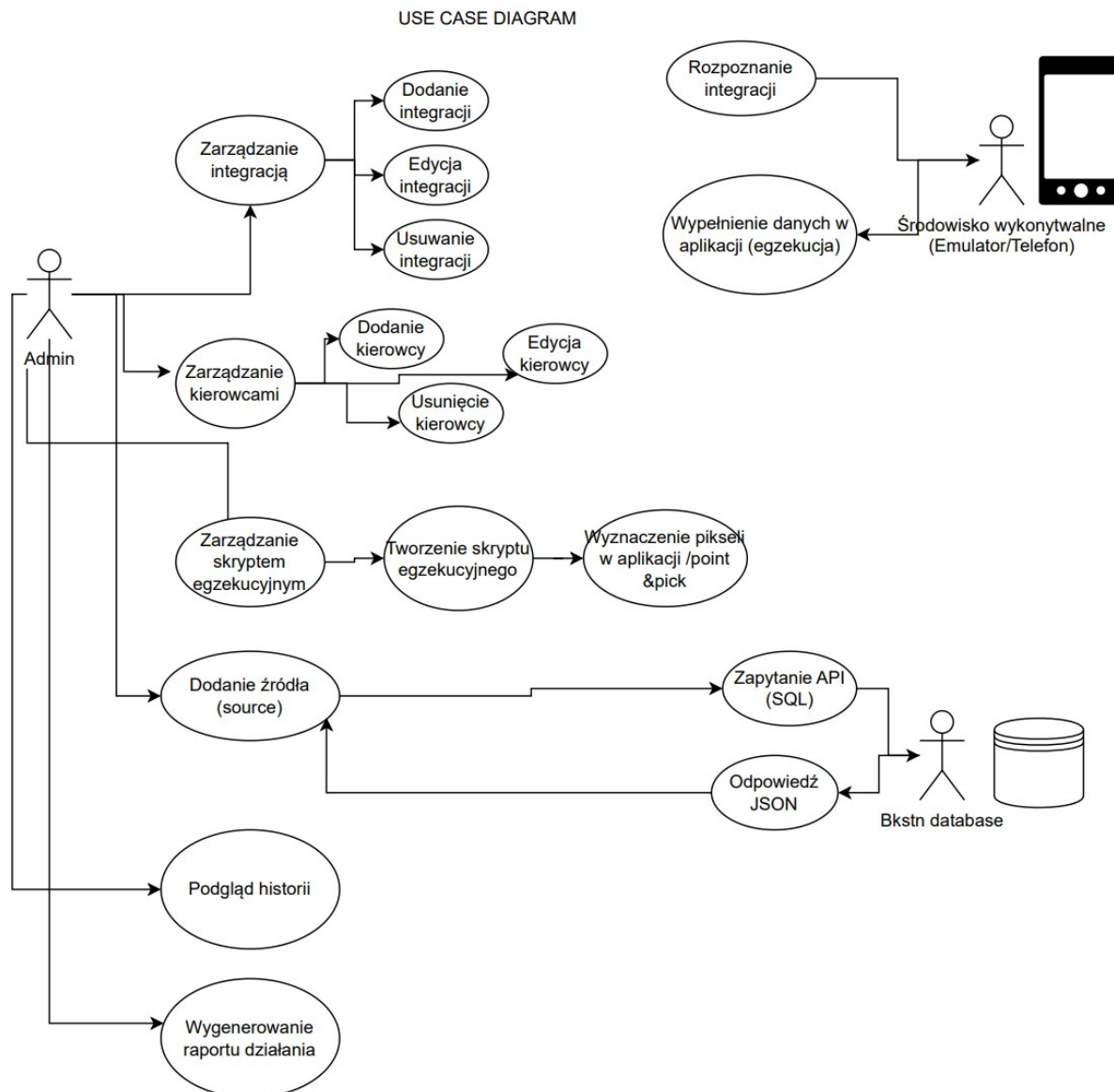


Diagram aktywności

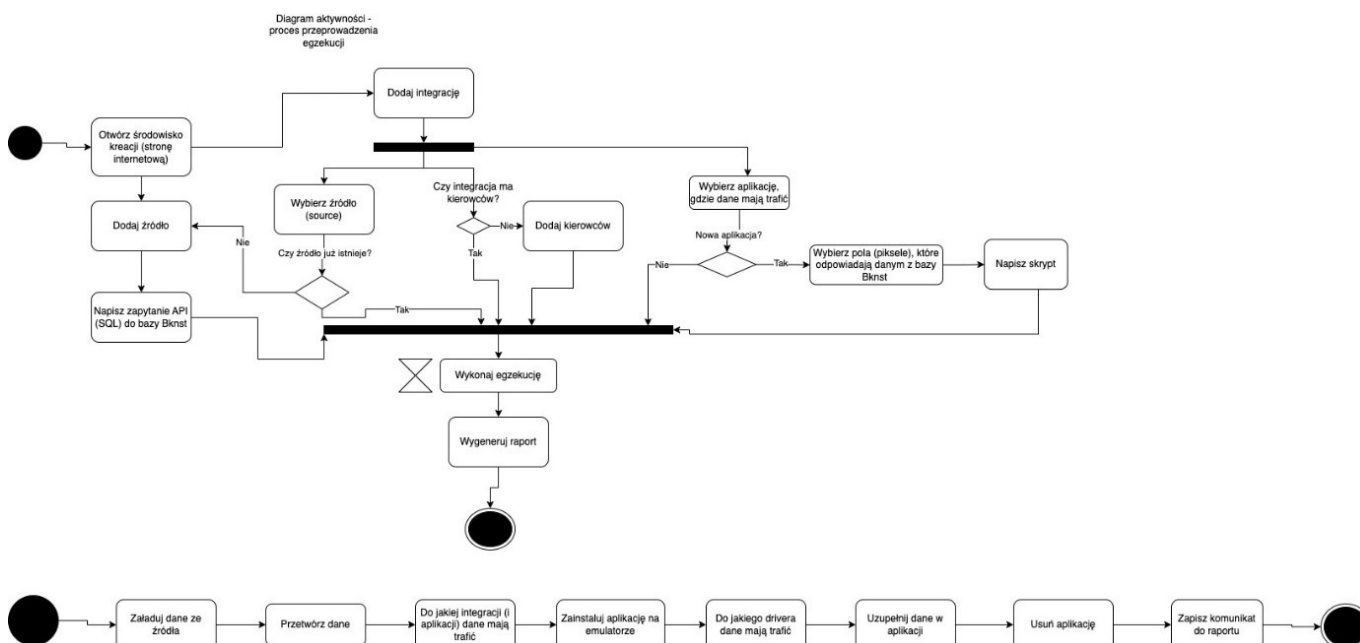
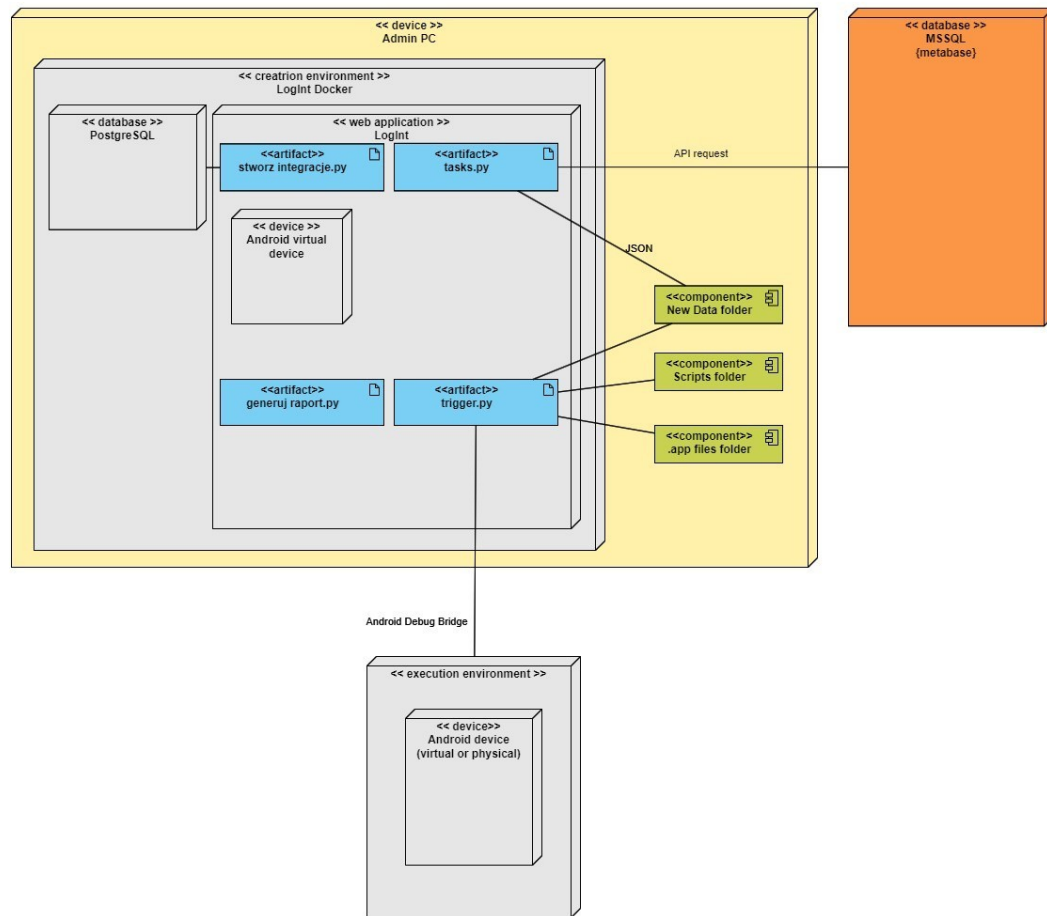
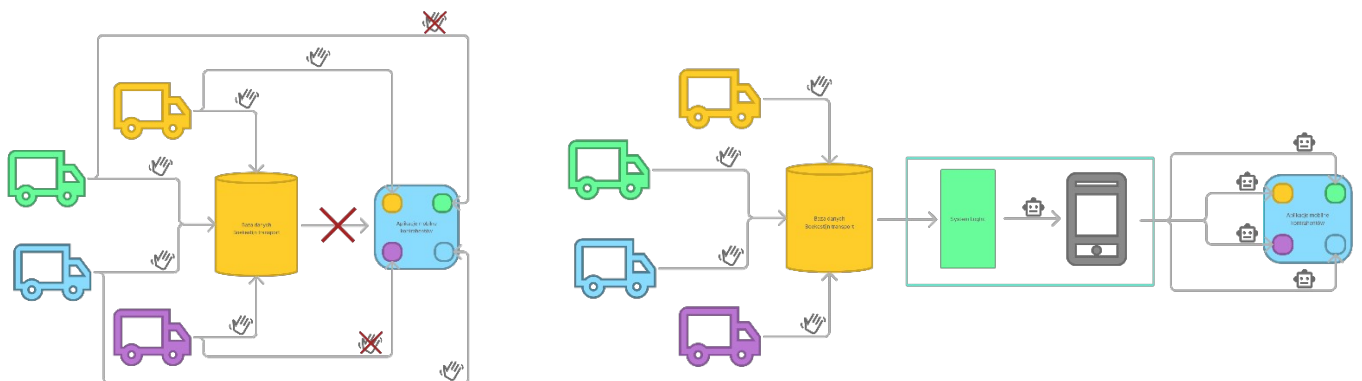


Diagram wdrożenia



Diagramy plakatowe:



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:
 - a. Zainstaluj Docker Engine
 - b. Zainstaluj Docker Compose
2. Instalacja Emulatora:

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

3. Instalacja systemu LogInt:
 - a. Sklonuj repozytorium LogInt
 - i. 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: LogIntProject/ConfigurationScripts
2. `chmod +x AndroidStudioSetup.sh`
3. `./AndroidStudioSetup.sh`

Android Studio można również zainstalować za pomocą Snap Store

Program AndroidStudio powinien zostać zainstalowany 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 powinien pokazać się na liście

2. Uruchom skrypt AvdSetup.sh
3. Zainstaluj Android SDK Command-line tools
 1. More Actions > SDK Manager > SDK Tools > Android SDK Command-line Tools
4. 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 terminala. Jest ono również niezbędne do egzekucji skryptów wykonywujące integracje. Aby go używać 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:
 - a. Przenieś się do katalogu roboczego systemu LogInt
 - i. cd LogIntProject
 - b. uruchom docker compose
 - i. docker-compose up -build
2. Uruchomienie środowiska egzekucji:
 - a. Uruchom program Android Studio
 - i. More actions > Virtual device manager
 1. Uruchom wcześniej przygotowany emulator o nazwie „executoner”

Po uruchomieniu obu środowisk emulator 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 użytkownika

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 edytować 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 możliwości przywrócenia
4. Add Driver
 - a. 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`

Dokumentacja Testowa

Dokumentacja testowa przedstawia przegląd testów przeprowadzonych w celu zapewnienia jakości i funkcjonalności systemu LogInt. Każda nowa funkcjonalność była testowana poprzez dodanie przypadków testowych, co umożliwiło wykrywanie i usuwanie błędów. Proces testowania obejmował zarówno testy jednostkowe, jak i integracyjne, zapewniając spójność oraz niezawodność. Kod został przetestowany w 96% według pakietu coverage.

Moduł History:

Testy jednostkowe:

1. Testowanie tworzenia historii:
 - Opis: Testuje poprawność tworzenia nowego rekordu historii.
 - Warunki początkowe: Brak rekordów historii w bazie danych.
 - Kroki testowe: Tworzenie nowego rekordu historii
 - Oczekiwane wyniki: Nowy rekord historii zostaje pomyślnie utworzony.
2. Testowanie edycji historii:
 - Opis: Testuje funkcjonalność edycji istniejącego rekordu historii.
 - Warunki początkowe: Istniejący rekord historii w bazie danych.
 - Kroki testowe: Edycja istniejącego rekordu historii za pomocą metody `save()`.
 - Oczekiwane wyniki: Dane w rekordzie historii zostają poprawnie zaktualizowane.
3. Testowanie usuwania historii:
 - Opis: Testuje możliwość usunięcia rekordu historii.
 - Warunki początkowe: Istniejący rekord historii w bazie danych.
 - Kroki testowe: Usunięcie rekordu historii za pomocą metody `delete()`.
 - Oczekiwane wyniki: Rekord historii zostaje pomyślnie usunięty z bazy danych.

Testy integracyjne:

1. Testowanie widoku historii:

- Opis: Testuje odpowiedź widoku historii.
 - Warunki początkowe: Brak rekordów historii w bazie danych.
 - Kroki testowe: Wyświetlenie widoku historii za pomocą klienta testowego.
 - Oczekiwane wyniki: Widok zwraca odpowiednią odpowiedź HTTP
2. Testowanie widoku pustej historii:
 - Opis: Testuje odpowiedź widoku historii w przypadku braku rekordów.
 - Warunki początkowe: Brak rekordów historii w bazie danych.
 - Kroki testowe: Wyświetlenie widoku historii za pomocą klienta testowego.
 - Oczekiwane wyniki: Widok wyświetla odpowiednią informację o braku rekordów

```
root@ad3fe055505b:/code/LogIntProject# python manage.py test history
Found 5 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
.....
-----
Ran 5 tests in 0.108s

OK
Destroying test database for alias 'default'...
```

Moduł Home:

Testy jednostkowe:

1. Testowanie dodawania nowej integracji:
 - Opis: Testuje możliwość dodania nowej integracji za pomocą formularza.
 - Warunki początkowe: Brak integracji w bazie danych.
 - Kroki testowe: Wypełnienie formularza dodawania integracji i przesłanie danych.
 - Oczekiwane wyniki: Nowa integracja zostaje pomyślnie dodana.
2. Testowanie wyświetlania szczegółów integracji:
 - Opis: Testuje dostępność i poprawne działanie widoku szczegółów integracji.
 - Warunki początkowe: Istniejąca integracja w bazie danych.
 - Kroki testowe: Wyświetlenie widoku szczegółów integracji za pomocą klienta testowego.
 - Oczekiwane wyniki: Widok zwraca odpowiednią odpowiedź HTTP i wyświetla szczegóły integracji.
3. Testowanie edycji integracji:
 - Opis: Testuje funkcjonalność edycji istniejącej integracji.
 - Warunki początkowe: Istniejąca integracja w bazie danych.
 - Kroki testowe: Edycja istniejącej integracji za pomocą formularza edycji.
 - Oczekiwane wyniki: Po zapisaniu zmian, integracja zostaje pomyślnie zaktualizowana.
4. Testowanie usuwania integracji:
 - Opis: Testuje możliwość usunięcia integracji.
 - Warunki początkowe: Istniejąca integracja w bazie danych.
 - Kroki testowe: Usunięcie integracji za pomocą przycisku usuń.
 - Oczekiwane wyniki: Integracja zostaje pomyślnie usunięta z bazy danych.

Testy integracyjne:

1. Testowanie dodawania konta kierowcy:
 - Opis: Testuje możliwość dodania nowego konta kierowcy do istniejącej integracji.
 - Warunki początkowe: Istniejąca integracja w bazie danych.
 - Kroki testowe: Dodanie nowego konta kierowcy do integracji za pomocą formularza.
 - Oczekiwane wyniki: Nowe konto kierowcy zostaje pomyślnie dodane do integracji.
2. Testowanie edycji konta kierowcy:
 - Opis: Testuje możliwość edycji danych konta kierowcy.
 - Warunki początkowe: Istniejące konto kierowcy w bazie danych.

- Kroki testowe: Edycja danych konta kierowcy za pomocą formularza edycji.
 - Oczekiwane wyniki: Po zapisaniu zmian, dane konta kierowcy zostają pomyślnie zaktualizowane.
3. Testowanie usuwania konta kierowcy:
 - Opis: Testuje możliwość usunięcia konta kierowcy z integracji.
 - Warunki początkowe: Istniejące konto kierowcy w bazie danych.
 - Kroki testowe: Usunięcie konta kierowcy z integracji za pomocą przycisku usuń.
 - Oczekiwane wyniki: Konto kierowcy zostaje pomyślnie usunięte z integracji.
 4. Testowanie usuwania integracji z poziomu strony głównej:
 - Opis: Testuje możliwość usunięcia integracji bezpośrednio z poziomu strony głównej.
 - Warunki początkowe: Istniejąca integracja wyświetlana na stronie głównej.
 - Kroki testowe: Usunięcie integracji za pomocą przycisku usuń
 - Oczekiwane wyniki: Integracja zostaje pomyślnie usunięta z bazy danych.
 5. Testowanie aktywacji integracji:
 - Opis: Sprawdza, czy integracja może być aktywowana.
 - Warunki początkowe: Nieaktywna integracja w bazie danych.
 - Kroki testowe: Aktywacja integracji za pomocą przycisku Activate.
 - Oczekiwane wyniki: Status integracji zmienia się na aktywny.
 6. Testowanie dezaktywacji integracji:
 - Opis: Sprawdza, czy integracja może być dezaktywowana.
 - Warunki początkowe: Aktywna integracja w bazie danych.
 - Kroki testowe: Dezaktywacja integracji za pomocą przycisku Deactivate.
 - Oczekiwane wyniki: Status integracji zmienia się na nieaktywny.

```

root@ad3fe055505b:/code/LogIntProject# python manage.py test home
Found 17 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
.....
-----
Ran 17 tests in 5.431s

OK
Destroying test database for alias 'default'...

```

Moduł Source:

Testy jednostkowe:

1. Testowanie tworzenia źródła:
 - a. Opis: Sprawdza, czy nowe źródło jest tworzone poprawnie.
 - b. Warunki początkowe: Brak nowego źródła w bazie danych.
 - c. Kroki testowe: Utworzenie nowego źródła za pomocą formularza.
 - d. Oczekiwane wyniki: Nowe źródło zostaje pomyślnie dodane do bazy danych.
2. Testowanie widoku źródeł:
 - a. Opis: Weryfikuje dostępność i poprawne działanie widoku źródeł.
 - b. Warunki początkowe: Istniejące źródła w bazie danych.
 - c. Kroki testowe: Wyświetlenie listy źródeł za pomocą klienta testowego.
 - d. Oczekiwane wyniki: Widok zwraca odpowiednią odpowiedź HTTP i wyświetla listę źródeł.
3. Testowanie dodawania nowego źródła:
 - a. Opis: Sprawdza możliwość dodania nowego źródła za pomocą formularza.
 - b. Warunki początkowe: Brak nowego źródła w bazie danych.
 - c. Kroki testowe: Wypełnienie formularza dodawania nowego źródła i przesłanie danych.
 - d. Oczekiwane wyniki: Nowe źródło zostaje pomyślnie dodane do bazy danych, a także zapisane w historii.
4. Testowanie edycji źródła:
 - a. Opis: Testuje funkcjonalność edycji istniejącego źródła.

- b. Warunki początkowe: Istniejące źródło w bazie danych.
 - c. Kroki testowe: Wybranie opcji edycji dla istniejącego źródła i zastosowanie zmian.
 - d. Oczekiwane wyniki: Dane źródła zostają zaktualizowane zgodnie z wprowadzonymi zmianami, a także zapisane w historii.
5. Testowanie edycji istniejącego źródła o tej samej nazwie:
- a. Opis: Sprawdza, czy błąd jest wyświetlany, gdy próbuje się edytować istniejące źródło, które ma tę samą nazwę co inne źródło.
 - b. Warunki początkowe: Istniejące źródło z tą samą nazwą w bazie danych.
 - c. Kroki testowe: Próba edycji istniejącego źródła, które ma tę samą nazwę co inne źródło.
 - d. Oczekiwane wyniki: Wyświetlenie odpowiedniego komunikatu o błędzie informującego o konflikcie nazw.
6. Testowanie edycji nieistniejącego źródła:
- a. Opis: Sprawdza, czy błąd jest wyświetlany, gdy próbuje się edytować nieistniejące źródło.
 - b. Warunki początkowe: Brak źródła o podanej nazwie w bazie danych.
 - c. Kroki testowe: Próba edycji nieistniejącego źródła.
 - d. Oczekiwane wyniki: Wyświetlenie odpowiedniego komunikatu o błędzie informującego o nieznalezieniu źródła.
7. Testowanie usuwania źródła:
- Opis: Sprawdza możliwość usunięcia źródła.
 - Warunki początkowe: Istniejące źródło w bazie danych.
 - Kroki testowe: Usunięcie źródła za pomocą odpowiedniej opcji.
 - Oczekiwane wyniki: Źródło zostaje pomyślnie usunięte z bazy danych, a także zapisane w historii.

Testy integracyjne:

1. Testowanie dodawania nowego źródła:
- Opis: Sprawdza możliwość dodania nowego źródła za pomocą formularza.
 - Warunki początkowe: Brak nowego źródła w bazie danych.
 - Kroki testowe: Wypełnienie formularza dodawania nowego źródła i przesłanie danych.
 - Oczekiwane wyniki: Nowe źródło zostaje pomyślnie dodane do bazy danych, a także zapisane w historii.
2. Testowanie edycji źródła:
- Opis: Testuje funkcjonalność edycji istniejącego źródła.
 - Warunki początkowe: Istniejące źródło do edycji.
 - Kroki testowe: Edycja istniejącego źródła poprzez formularz edycji.
 - Oczekiwane wyniki: Poprawna aktualizacja danych źródła i zapisanie odpowiedniej historii.
3. Testowanie edycji istniejącego źródła o tej samej nazwie:
- Opis: Sprawdza, czy błąd jest wyświetlany, gdy próbuje się edytować istniejące źródło, które ma tę samą nazwę co inne źródło.
 - Warunki początkowe: Istniejące źródło o tej samej nazwie co inne źródło.
 - Kroki testowe: Próba edycji istniejącego źródła poprzez formularz edycji.
 - Oczekiwane wyniki: Wyświetlenie komunikatu o błędzie informującego o konflikcie nazw.
4. Testowanie edycji nieistniejącego źródła:
- Opis: Sprawdza, czy błąd jest wyświetlany, gdy próbuje się edytować nieistniejące źródło.
 - Warunki początkowe: Brak istniejącego źródła o podanej nazwie.
 - Kroki testowe: Próba edycji nieistniejącego źródła poprzez formularz edycji.

- Oczekiwane wyniki: Wyświetlenie komunikatu o błędzie informującego o braku istnienia źródła.

5. Testowanie usuwania źródła:

- Opis: Sprawdza możliwość usunięcia źródła.
- Warunki początkowe: Istniejące źródło do usunięcia.
- Kroki testowe: Wywołanie funkcji usuwania źródła.
- Oczekiwane wyniki: Potwierdzenie usunięcia źródła i zapisanie odpowiedniej historii

```
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
Test Source
.....New Source
.....
-----
Ran 10 tests in 0.165s

OK
Destroying test database for alias 'default'...
```

Moduł Reports:

Testy jednostkowe:

1. Testowanie generowania pliku PDF:

- Opis: Weryfikuje, czy plik PDF jest generowany poprawnie.
- Warunki początkowe: Brak konkretnych warunków początkowych.
- Kroki testowe: Wywołanie funkcji generującej plik PDF na przykładowych danych.
- Oczekiwane wyniki: Po wykonaniu funkcji generującej plik PDF oczekiwany jest plik PDF o niezerowej wielkości.

2. Testowanie generowania pliku tekstowego:

- Opis: Weryfikuje, czy plik tekstowy jest generowany poprawnie.

- Warunki początkowe: Brak konkretnych warunków początkowych.
- Kroki testowe: Wywołanie funkcji generującej plik tekstowy na przykładowych danych.
- Oczekiwane wyniki: Po wykonaniu funkcji generującej plik tekstowy oczekiwany jest plik tekstowy o niezerowej wielkości.

3. Testowanie pobierania wszystkich raportów:

- Opis: Sprawdza, czy można pobrać wszystkie raporty w formacie PDF.
- Warunki początkowe: Istnieją raporty do pobrania.
- Kroki testowe: Wywołanie funkcji pobierającej wszystkie raporty.
- Oczekiwane wyniki: Po wykonaniu funkcji pobierającej raporty oczekiwane są pliki PDF z danymi raportów.

4. Testy integracyjne: Brak testów integracyjnych dla modułu reports.

```
root@ad3fe055505b:/code/LogIntProject# python manage.py test reports
Found 3 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
...
-----
Ran 3 tests in 0.083s

OK
Destroying test database for alias 'default'...
```


Pokrycie kodu przez testy

root@8c4c655f5e99:/code/LogIntProject# coverage report -m				
Name	Stmts	Miss	Cover	Missing

LogInt/__init__.py	3	0	100%	
LogInt/celery.py	8	0	100%	
LogInt/settings.py	26	0	100%	
LogInt/urls.py	5	0	100%	
history/__init__.py	0	0	100%	
history/admin.py	7	0	100%	
history/apps.py	4	0	100%	
history/migrations/0001_initial.py	5	0	100%	
history/migrations/0002_alter_history_operation_date.py	4	0	100%	
history/migrations/__init__.py	0	0	100%	
history/models.py	8	0	100%	
history/tests.py	53	0	100%	
history/urls.py	3	0	100%	
history/views.py	11	0	100%	
home/__init__.py	0	0	100%	
home/admin.py	9	0	100%	
home/apps.py	4	0	100%	
home/choices.py	2	0	100%	
home/management/__init__.py	0	0	100%	
home/management/commands/__init__.py	0	0	100%	
home/migrations/0001_initial.py	6	0	100%	
home/migrations/0002_alter_integration_source.py	5	0	100%	
home/migrations/0003_remove_integration_account_id_and_more.py	4	0	100%	
home/migrations/0004_alter_integration_source.py	5	0	100%	
home/migrations/__init__.py	0	0	100%	
home/models.py	22	0	100%	
home/tests.py	205	0	100%	
home/urls.py	4	0	100%	
home/views.py	146	15	90%	214-234
manage.py	11	2	82%	12-13
pages/__init__.py	0	0	100%	
pages/admin.py	1	0	100%	
pages/apps.py	4	0	100%	
pages/migrations/__init__.py	0	0	100%	
pages/models.py	1	0	100%	
pages/tests.py	11	0	100%	
pages/urls.py	3	0	100%	
pages/views.py	3	0	100%	
reports/__init__.py	0	0	100%	
reports/admin.py	1	0	100%	
reports/apps.py	4	0	100%	
reports/migrations/__init__.py	0	0	100%	
reports/models.py	1	0	100%	
reports/tests.py	66	0	100%	
reports/urls.py	3	0	100%	
reports/views.py	92	0	100%	
sources/__init__.py	0	0	100%	
sources/admin.py	5	0	100%	
sources/apps.py	4	0	100%	
sources/migrations/0001_initial.py	5	0	100%	
sources/migrations/0002_remove_source_id_alter_source_source_name.py	4	0	100%	
sources/migrations/__init__.py	0	0	100%	
sources/models.py	6	0	100%	
sources/tests.py	88	0	100%	
sources/urls.py	3	0	100%	
sources/views.py	47	1	98%	20
steps/__init__.py	0	0	100%	
steps/admin.py	5	0	100%	
steps/apps.py	4	0	100%	
steps/migrations/0001_initial.py	6	0	100%	
steps/migrations/0002_alter_steps_action.py	4	0	100%	
steps/migrations/0003_alter_steps_img.py	4	0	100%	
steps/migrations/0004_alter_steps_img.py	4	0	100%	
steps/migrations/__init__.py	0	0	100%	
steps/models.py	20	0	100%	
steps/tests.py	92	0	100%	
steps/views.py	74	23	69%	29, 56-58, 75-87, 92-98

TOTAL	1125	41	96%	