|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | Politechnika Śląska  **Hurtownie Danych i Systemy Eksploracji Danych** | |  | |
| **Rok akademicki** | | **Rodzaj studiów\*: SSI/NSI/NSM** | **Semestr :** | **Grupa** | **Sekcja** |
| **2019/2020** | | **SSI** | **6** | **BDiS** | **7** |
| **Data i godzina planowana ćwiczenia:** | | **Poniedziałek - 17:30** | **Przedmiot** : | **HDISED** | |
| **Prowadzący :** | | **Marcin Gorawski** |
| **Raport z projektu zasadniczego** | | | | | |
| **Temat ćwiczenia:**  **Rozproszony system ekstrakcji danych z Internetu Rzeczy** | | | | | |
| **Skład sekcji:** | 1. Wojciech Krupowies   2. Mateusz Kula | | | | |

1. Opis projektu

Projekt jaki postanowiliśmy zrealizować w ramach przedmiotu to dwie aplikacje mobilne, jedna na urządzenia iOS, a druga na urządzenia z systemem Android. Obie aplikacje wykorzystują czujniki w które wyposażone są obecne telefony, a następnie zebrane informacje gromadzone są w bazie danych. Jedna z aplikacji wykorzystuje krokomierz, a druga wbudowany w telefon czujnik natężenia światła. Zebrane informacje następnie są gromadzone w bazie danych.

1. Analiza, projektowanie

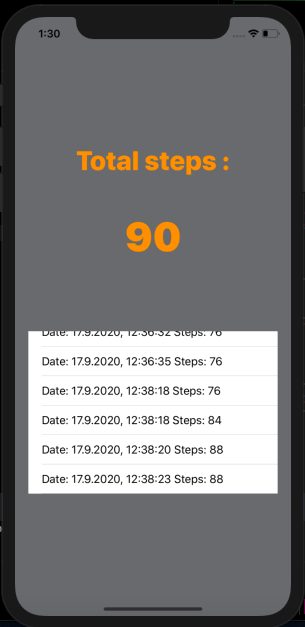
Przed przystąpieniem do pracy podzieliliśmy realizacja projektu na dwa problemy. Pierwszy z nich to implementacja dostępu do sensorów oraz odczytywanie interesujących nas wartości. W przypadku aplikacji iOS z pomocą przyszedł tutaj moduł CoreMotion, który umożliwia pobieranie danych z czujników takich jak akcelerometr, żyroskop czy też krokomierz. W przypadku aplikacji Android SensorManager do obsługi czujnika natężenia światła, FusedLocationClient do pobrania lokalizacji oraz OkHttp do obsługi połączeń sieciowych.

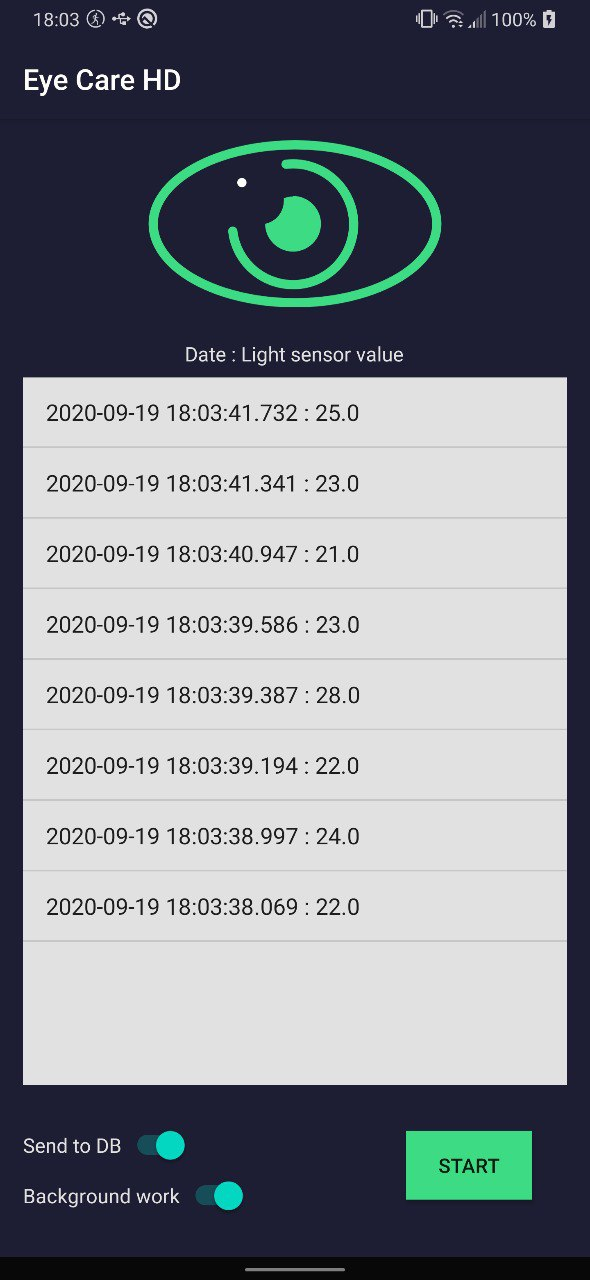
Kolejnym aspektem projektu było gromadzenie danych. Tutaj dla danych pobranych z krokomierza zdecydowaliśmy się na bardzo wygodne, praktyczne i nowoczesne rozwiązanie jakim jest platforma Firebase zaproponowana przez Google. Dane gromadzone z aplikacji Android posiadają nasze własne rozwiązanie oparte o:

* bazę MSSQL uruchomioną jako kontener Dockera na serwerze z Linuxem
* serwis REST-owy w technologii .NET CORE, których może być uruchomiony na Windowsie, Linuxie jak i MacOS, co pozwala utworzyć wiele endpointów dla aplikacji mobilnych

1. Specyfikacja zewnętrzna

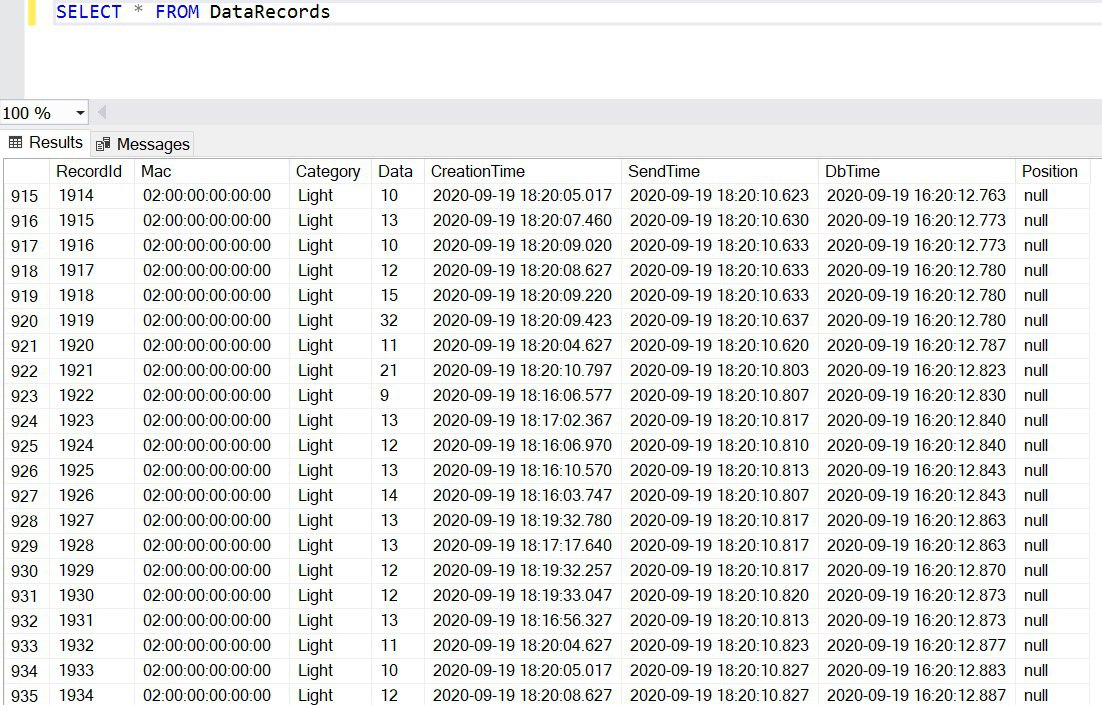
Aplikacje składają się z jednego widoku na którym prezentujemy aktualną wartość pobraną z czujnika oraz tabele, która prezentuje poprzednie wartości.



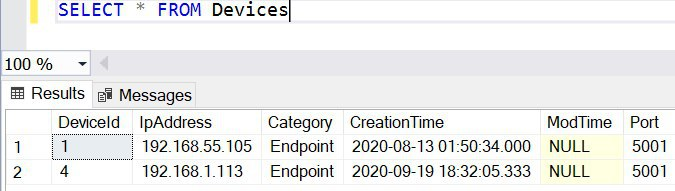


Widok aplikacji Android

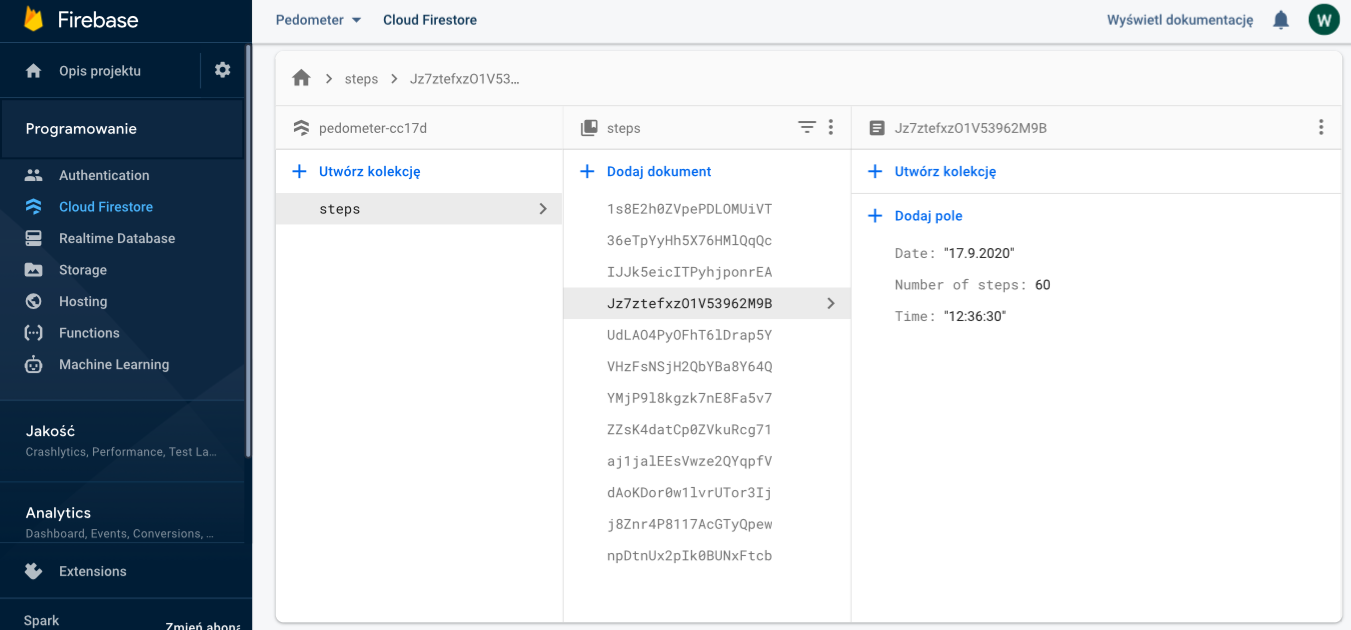
Widok aplikacji iOS



Tabele bazy MSSQL dla aplikacji Android



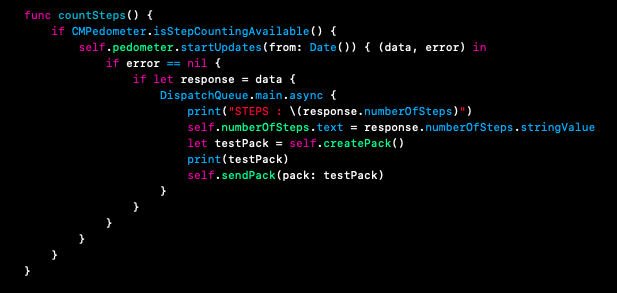
Firebase

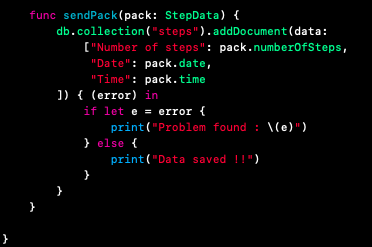
Wykorzystując platformę Firebase możemy podglądać nasze dane w przeglądarce internetowej.

1. Specyfikacja wewnętrzna

**Aplikacja iOS**

W przypadku programowania aplikacji mobilnych na urządzenia z systemem iOS zostaje udostępniony nam wyżej wspomniany moduł CoreMotion. Dzięki niemu możemy w bardzo wygodny sposób możemy odczytać wartość z krokomierza, który jest wbudowany w naszych telefonach.

Poniżej prezentuje dwie najważniejsze metody : ***countSteps,*** która jest odpowiedzialna za pobranie informacji z sensora, funkcja ta pobiera dane za każdym razem gdy sensor wyczuje zmianę oraz funkcja ***sendPack,*** która jest odpowiedzialna za wysłanie „paczki” do bazy danych. Paczka to zaimplementowana struktura ***StepData***, która przechowuje ilość kroków oraz datę i godzinę odczytania wartości z czujnika. 



Funkcja ***countSteps*** odpowiedzialna za pobranie informacji z sensora

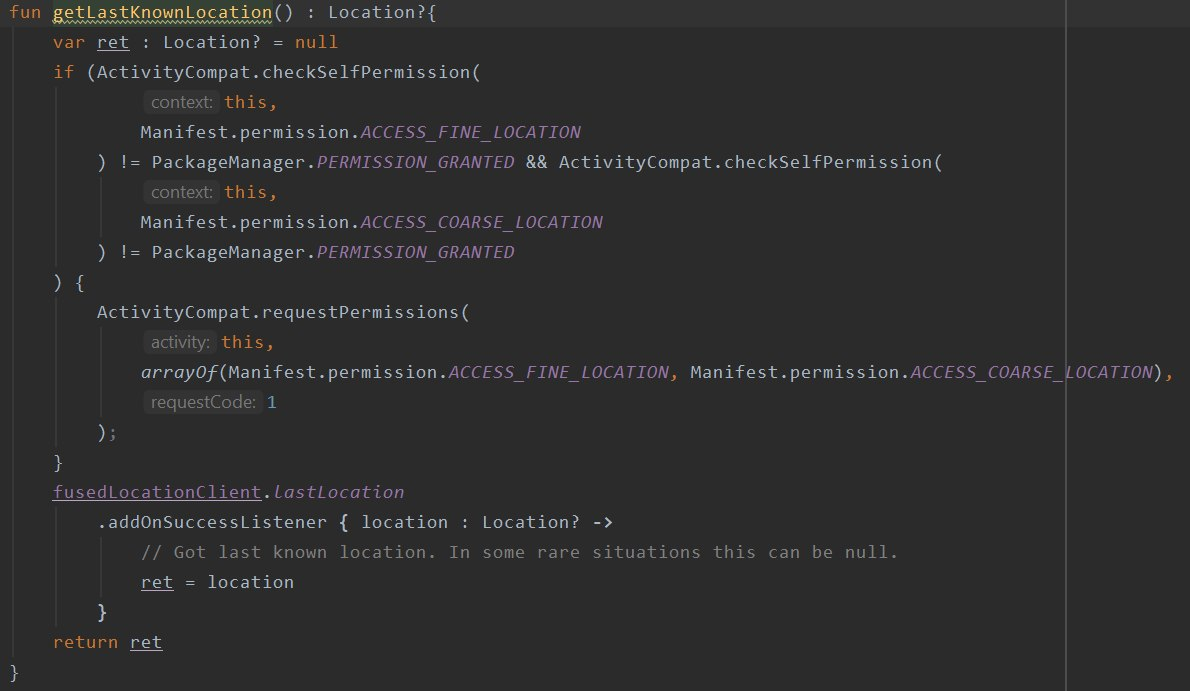
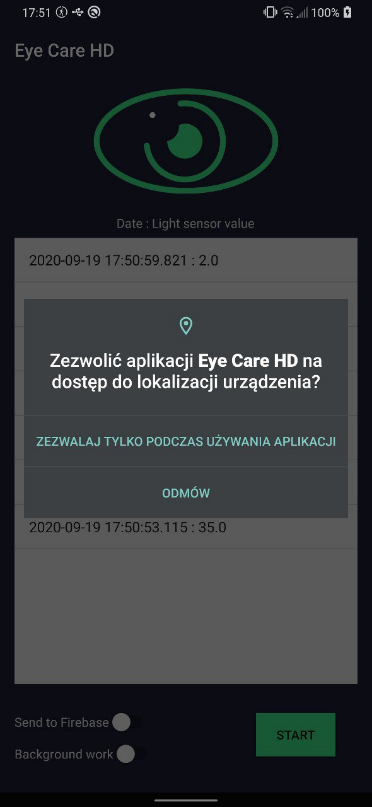
Funkcja ***sendPack*** odpowiedzialna za wysyłanie danych do bazy.

**Aplikacja Android**

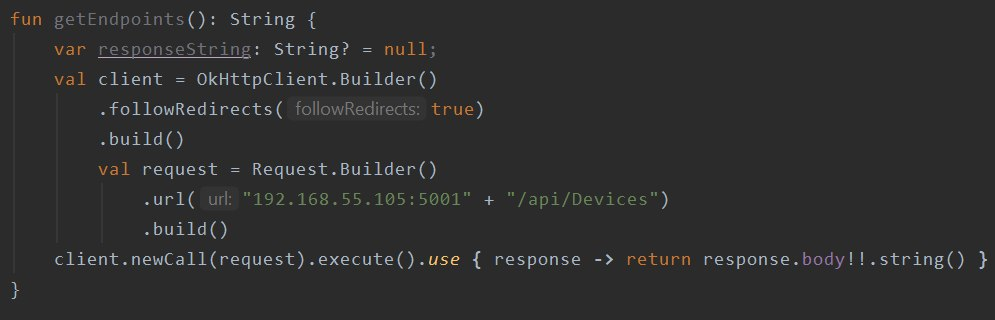
Jest utworzona w bardzo przystępnym języku Kotlin (następca Javy w aplikacjach na platformę Android). Najważniejszymi metodami są odczyty z sensora natężenia światła:

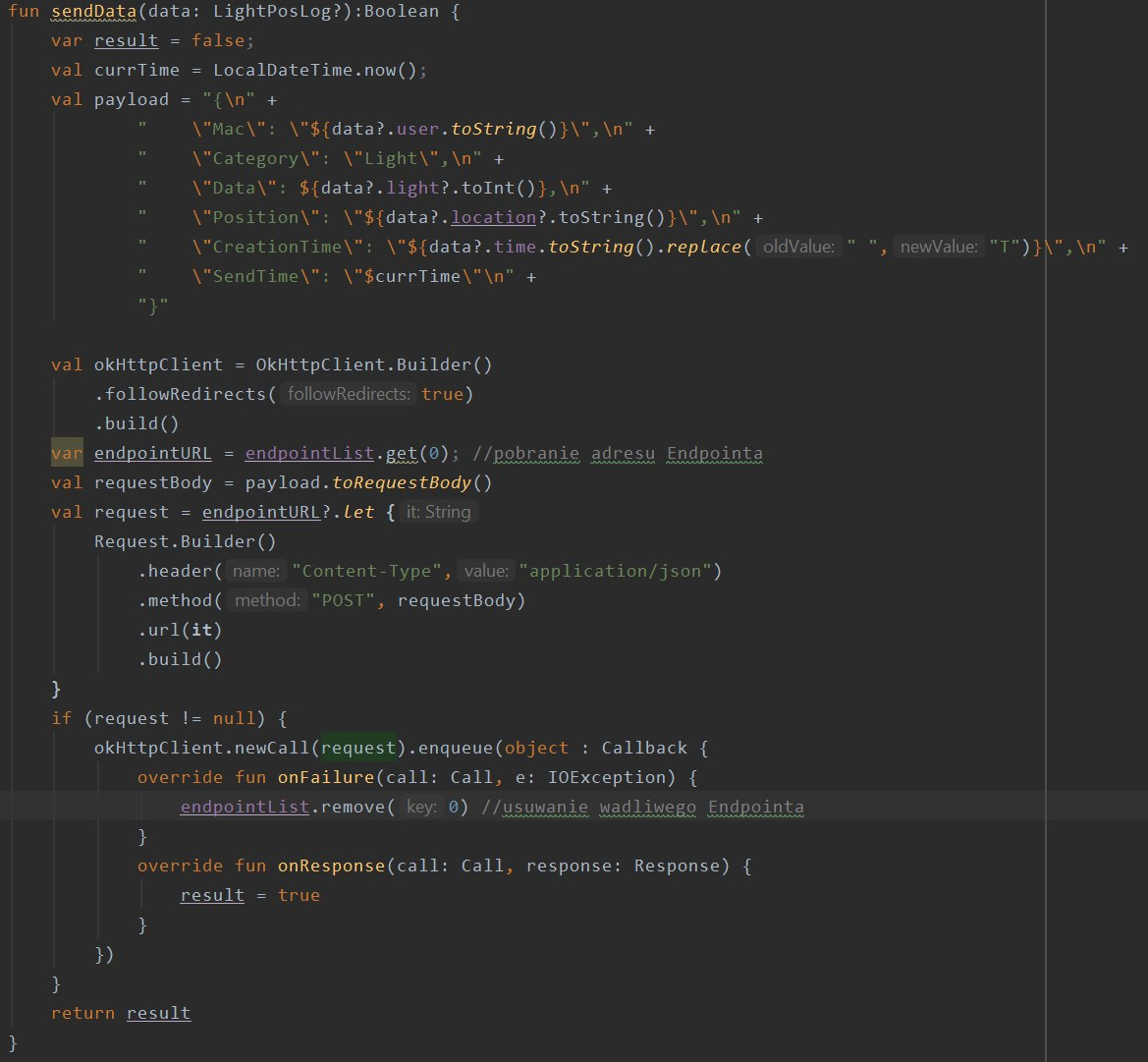


Za pozyskiwanie danych o lokalizacji urządzenia odpowiada metoda:

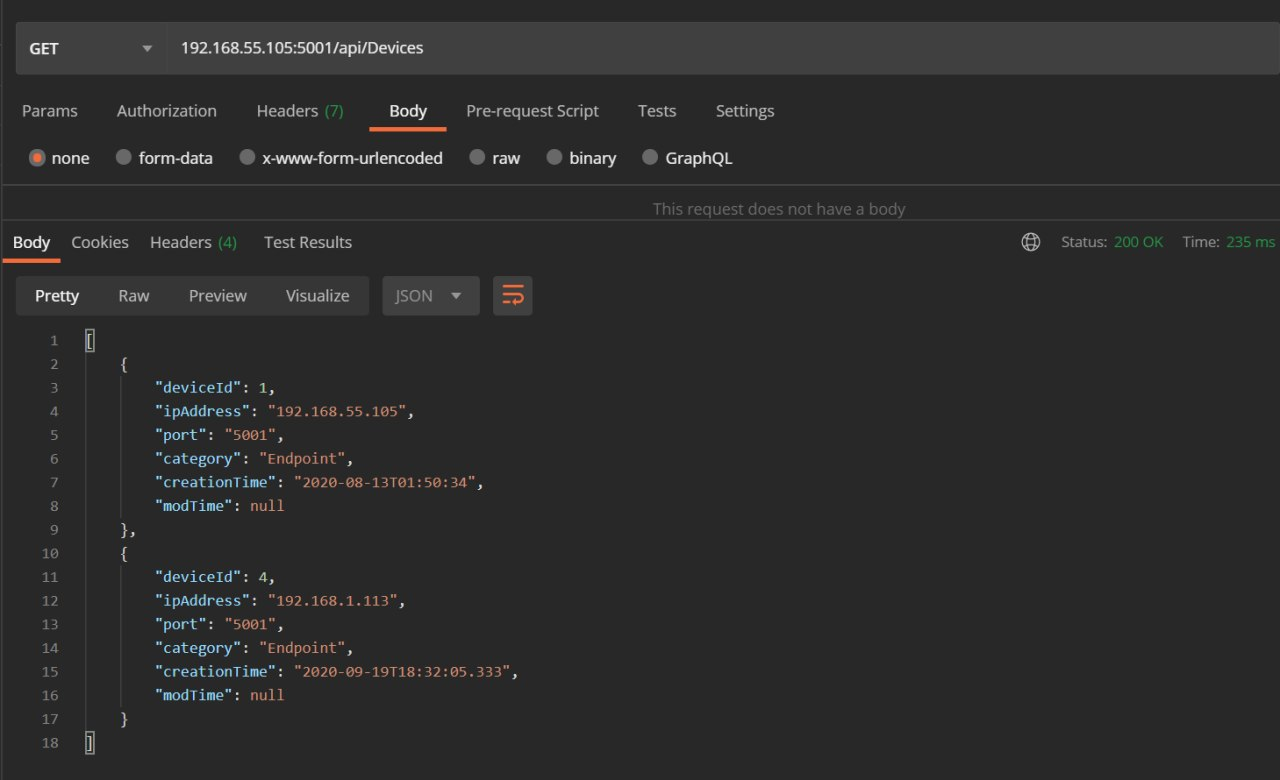
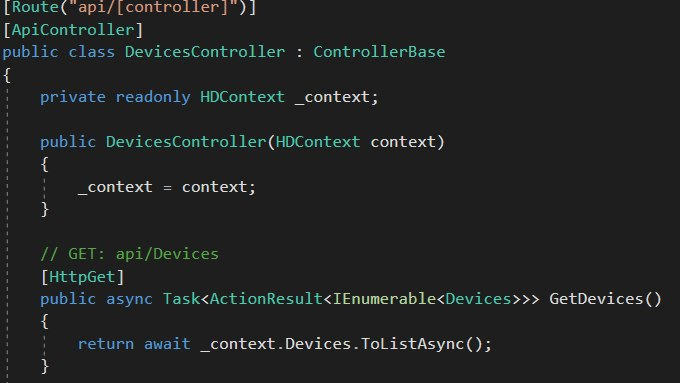


Dodatkowo ważnym aspektem jest pozyskanie zgody na dostęp aplikacji do funkcji telefonu takich jak czujnik światła, lokalizacja oraz dostęp do internetu.

Dostępne endpointy pobierane są przy użyciu metody GET dla 

Wysyłanie danych odbywa się przy użyciu funkcji, w której dodatkowo jest mechanizm wykrywania wadliwego połączenia: 

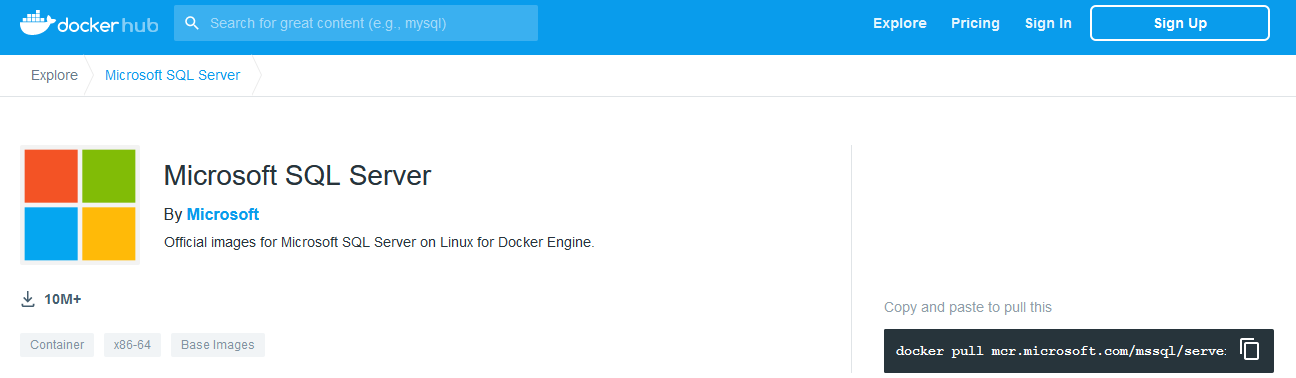
**Serwis REST**

Posiada model danych utworzony przy użyciu biblioteki Microsoft .NET CORE Entity Framework na podstawie wcześniej utworzonej bazy MSSQL na zdalnym serwerze. Do obsługi zapytań RESTowych został użyty ASP.NET, zostały utworzone dwa Controllery które posiadają komplet metod http (GET, POST, PUT, DELETE). Dzięki wykorzystaniu technologii .NET Core projekt może być skompliowany na wiele systemów operacyjnych czyniąc go bardzo wszechstronnym. 

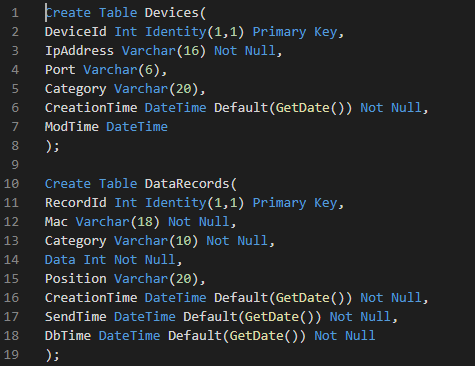
Przykładowy kod instrukcji GET oraz jego odpowiedź pobrana z bazy danych.

**Baza MSSQL**

Została utworzona na maszynie działającej pod kontrolą systemu Linux w kontenerze Dockera z obrazu udostępnionego developerom. Wybór na tą bazę padł dzięki nauce wynikającej z wykorzystywania w projekcie wstepnym przedmiotu HDISED co przyzwyczaiło nas do korzystania z Microsoft SQL Server Management Studio.



Skrypty tworzące bazę:



1. Testowanie

W celu przetestowania aplikacji zostały one zainstalowane na urządzeniach fizycznych. Programy działają dobrze i spełniają swoją rolę. Dane są wysyłane do bazy pomyślnie. Testowana również była sytuacja w której telefon tracił połączenie z bazą danych podczas działania aplikacji. Korzystając z platformy Firebase scenariusz ten jest już zabezpieczony i dane nie zostają utracone, przy ponownym połączeniu zaległe „paczki” zostają przesłane do bazy danych. W aplikacji Android po utraceniu połączenia z serwisem następuje próba połączenia z inną instancją serwisu pod innym adresem, przy pozytywnym połączeniu dane, które nie zostały wysłane zostają wysłane na bazę. Wnioski

Mimo panującego zamieszania udało się sprawnie wykonać projekt. Cieszy nas fakt, że mogliśmy go realizować w wybranym przez nas języku programowania oraz tworzyć aplikacje w interesującej nas technologii. Aplikacje działają sprawnie i realizują podstawowe założenia projektu pozostawiając wiele miejsca to rozwoju projektu o analizę lub porównywanie danych.

Dodatkowo po pozyskaniu publicznego adresu IP od dostawcy internetowego możliwe byłoby wystawienie endpointów „na świat” umożliwiając działanie poza siecią domową.

Dzięki realizacji tego projektu mogliśmy zgłębić naszą wiedzę na temat programowania urządzeń mobilnych oraz zmierzyć się z zagadnieniem ekstrakcji danych. Wybrane przez nas narzędzia okazały się wygodne w użyciu oraz dobrze spełniły początkowe założenia projektu.