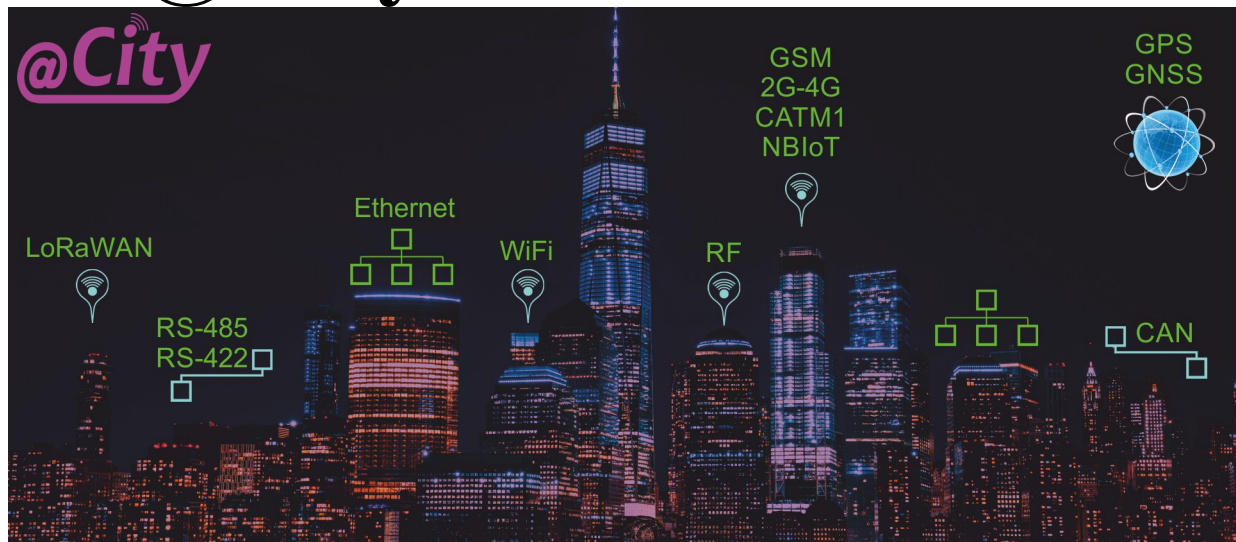


# @City Platforma IoT



iSys – Intelligent Systems

**@City**

**@AirQ**

**@Bin**

**@Light @Metering @Trace**

iSys.PL

**Confidential Information**

## Table of Contents

1. Wstęp.....	5
1.1 Obsługiwane typy urządzeń.....	5
1.2. Obsługiwane typy produktów.....	5
1.3. Obsługiwane protokoły komunikacyjne.....	5
1.4. Obsługiwane łącza transmisyjne z urządzeń.....	5
1.5. @City Serwer Chmury.....	6
1.5.1. Serwer @City i bramki komunikacyjne TCP/UDP.....	7
1.5.2. Integracja HTTP LoRaWAN.....	7
1.5.3. Interfejs Frond-end .....	8
1.5.3. Uprawnienia Serwera @City.....	8
1.6. Inteligentne urządzenia IoT, CIoT.....	9
1.6.1. CIoT – urządzenia GSM .....	9
1.6.2. IoT – urządzenia LoRaWAN.....	9
1.6.2. BAS, BMS, IoT – urządzenia Ethernet, WiFi .....	9
1.7. Warianty współpracy Business to Business (B2B) .....	9
2. Funkcjonalność platformy IoT @City .....	11
3. Strona Główna.....	12
4. Formularz główny.....	12
4.1. Nagłówek.....	13
4.1.1. Home Link - (otwiera aktualną tabelę).....	13
4.1.2. 'X' pole – otwiera/zamyka formularz wyszukiwania I opcji.....	13
4.1.3. 'V' pole – otwiera/zamyka Formularz filtra pól.....	13
4.1.4. Ikony - linki do rezultatów wizualizacji (edytowalne).....	13
4.2. Formularz:.....	13
4.2.1. “X” pole - otwiera/zamyka cały formularz wyszukiwania I opcji.....	13
4.2.2. “CSS” – Wybór motywu wizualizacji.....	13
4.2.3.”Visible Fields” – pokazuje/ukrywa formularz filtra pól.....	13
4.2.4. “Tab” - Nazwa zakładki do dodawania I usuwania.....	13
4.2.5. “Add / Remove” przyciski – Dodaje I usuwa zakładki z nazwą znajdującą się w polu Tab .....	13
4.2.6. “Select Core” - główne pola.....	13
4.2.7. “Deselect All” - odznaczyć wszystkie pola.....	13
4.2.7. “Select All” - wybierz wszystkie pola.....	13
4.2.8. “Hide Filter” – Ukryj formularz.....	13
4.2.9. “Execute” – uruchomienie zmiany parametrów.....	14
4.2.10. 'V' pole – pokazuje/ukrywa Filtr Pól.....	14
4.3. Zakładki.....	14
4.4. Zawartość tabeli.....	14
4.4.1. Run – typ widoku rezultatów.....	14
4.4.2. Copy (+/- linki).....	14
4.4.3. Linki w komórkach tabeli.....	14

4.5. Kolejność danych.....	14
4.6. Przykład.....	15
5. Mapy.....	16
5.1. Inicjalizacja Mapy.....	16
5.2. Dodatkowe ustawienia zapytań.....	16
5.2.1. Zmiana skali Mapy (Zoom Level).....	17
5.2.2. IMEI (Wybór urządzenia).....	17
5.2.3. Lon, Lat (Długość, szerokość geograficzna - współrzędne).....	17
5.2.4. Modyfikacja Stylu Mapy (Motywu).....	17
5.2.5. Klauzula WHERE .....	17
5.2.6. Execute (Uruchamia zapytanie).....	17
5.2.7. Deselect All (Wyłącza wszystkie pola filtra).....	18
5.2.8. 'V' (Otwórz/zamknij Formularz Filtra).....	18
5.2.9. 'X' (Pokaż/ukryj formularz zapytania I opcji).....	18
5.3. Przykład.....	18
6. Wizualizacja wyników w tabeli.....	20
6.1. Inicjalizacja tabeli.....	20
6.2. Opcjonalne ustawienia zapytania.....	21
6.2.1. Sort - sortowanie pola rosnąco/malejąco.....	21
6.2.2. DB / IMEI – Wybór urządzenia/tabeli.....	21
6.2.3. CSS – wybierz styl (motyw wizualizacji).....	22
6.2.4. “Visible Fields” – Pokaż/zamknij formularz filtra pól.....	22
6.2.5. “Remove Empty” – Nie pokazuje pustych kolumn.....	22
6.2.6. 'X' pol (pokaż/zamknij formularz zapytań).....	22
6.2.7. Klauzula Where (ograniczanie danych).....	22
6.2.8. “Select Core” (Włącz większość popularnych pól).....	22
6.2.9. “Deselect All” (Usuń wszystkie pola z zapytania).....	22
6.2.10. “Execute” (Uruchom zapytanie).....	22
6.2.11. 'V' pole (otwórz/zamknij formularz filtra pól).....	22
7. Wykresy Słupkowe. ....	23
8. Wykresy Archiwalne. ....	24
8.1. Inicjalizacja wykresów archiwalnych.....	24
8.2. Opcjonalne ustawienia dla wykresów historycznych.....	25
8.2.1. IMEI – (Wybór urządzenia do wyświetlenia danych).....	25
8.2.2. “Min” - ograniczyć minimalną wartość pierwszego pola.....	25
8.2.3. “Max” - ograniczyć maksymalną wartość pierwszego pola.....	25
8.2.4. 'V' – Pokaż/ukryj formularz filtra pól.....	25
8.2.5. “From” ustaw minimalną datę/godzinę daty (*).....	25
8.2.6. “To” ustaw maksymalną datę/godzinę daty (*).....	25
8.2.7. 'X' pole (Pokaż/ukryj formularz zapytań).....	25
8.2.8. Klauzula “Where” .....	25
8.2.9. “Deselect All” (Usuń wszystkie pola z zapytania).....	25
8.2.10. “Execute” (uruchom zapytanie).....	26
8.2.11. 'V' Pole wyboru (otwórz/zamknij formularz filtra pól).....	26

8.3. Wykres Słupkowy : (wyświetla tylko istniejące dane).....	26
8.4. Wykres ciągły: (dla tych samych danych wejściowych):.....	27
9. Zgodność z przeglądarką internetową.....	27
10. Kustomizacja widoku/motywu.....	28
11. Tworzenie indywidualnych algorytmów przetwarzania danych.....	29
12. Struktura Bazy Danych.....	30
12.1. 'Struktura tabel ithings_' i '*' .....	31
12.2. Kolejka komend sterujących dla urządzeń (Events) '*_c' .....	32
12.3. Uzyskiwanie dostępu do wyników z baz danych - Mid-Level (odczyt danych).....	32
12.3.1. Odczyt aktualnych statusów wszystkich sterowników.....	32
12.3.2. Odczyt danych archiwalnych z urządzenia.....	33
12.3.3. Odczyt pojedynczego pola z aktualnych statusów z ograniczeniem wyników dla wszystkich urządzeń.....	34

## 1. Wstęp.

Platforma @City to dedykowany system „mini-chmury” dla indywidualnych klientów B2B. Platforma nie jest dzielona między użytkowników i tylko jeden klient ma dostęp do serwera fizycznego lub wirtualnego (VPS lub serwery dedykowane). Klient może wybrać jeden z kilkudziesięciu centrów danych w Europie lub na świecie.

### 1.1 Obsługiwane typy urządzeń.

Platforma @City jest dedykowana produktom iSys.PL

- IoT – RF/LoRaWAN (Internet of Things)
- CIoT – 2G/3G/4G/CATM1/NB IoT (Cell Internet of Things)
- WiFi
- Ethernet

### 1.2. Obsługiwane typy produktów.

Chmury @City (eCity) są dedykowane urządzeniom IoT pracującym w sieciach IP:

- @City
- @Light
- @Metering
- @Trace
- @AirQ
- @Bin

### 1.3. Obsługiwane protokoły komunikacyjne

Platforma IoT @City obsługuje następujące protokoły komunikacyjne:

- UDP - sugerowany dla czujników / urządzeń CIoT (zwłaszcza NB IoT) - najniższe wykorzystanie danych
- TCP IP - sugerowany dla urządzeń z komunikacją dwukierunkową – z potwierdzeniem i zdalnym sterowaniem
- HTTP - sugerowany tylko dla dostępu do danych, wizualizacji, eksportu, komunikacji „chmura – chmura”
- HTTP Webhook – dla LoRaWAN (integracja serwera sieciowego + aplikacyjnego LoRaWAN z chmurą @City)

### 1.4. Obsługiwane łącza transmisyjne z urządzeń

Platforma IoT @City obsługuje:

- Urządzenia GSM: 2G, 3G, 4G (LTE), CATM1 (LTEM1), NBIoT (komunikacja UDP / TCPIP)
- Sterowniki Ethernet (komunikacja UDP / TCP)
- Sterowniki WiFi (komunikacja UDP / TCP)
- *Przyszłe produkty pracujące w sieciach IP*
- Sterowniki LoRaWAN (za pośrednictwem Bramki LoRaWAN oraz serwera sieciowego+aplikacyjnego)
- Inne podsystemy nie pracujące w sieci IP, za pośrednictwem dodatkowej lokalnej bramy sprzętowej @City / eHouse.PRO

**Platforma IoT @City jest dedykowana:**

- Oryginalne produkty iSys
  - Urządzenia CIoT (pracujące w sieciach GSM)
  - Urządzenia IoT (pracujące w sieciach LoRaWAN)
  - WiFi - dedykowane produkty dla platformy @City
  - Ethernet - dedykowane produkty dla platformy @City
- Produkty pochodzące z koprodukcji (produkowane przez firmy trzecie w ramach licencji iSys pod zewnętrzną marką)
- Produkty franczyzowe (produkowane przez firmy trzecie na licencji iSys pod markami iSys)

## 1.5. @City Serwer Chmury

Oprogramowanie @City działa na serwerach VPS pracujących na systemie Linux (Virtual Private Server) lub serwerze dedykowanym po stronie internetowej, w zależności od żądanej wydajności serwera (zwanej dalej **serwerem**). Wymagana wydajność zależy od następujących czynników:

- dostęp prywatny / publiczny
- łączna liczba urządzeń
- częstotliwość aktualizacji statusu urządzeń
- częstotliwość odświeżania danych

**Istnieje kilka możliwych wariantów serwerów VPS w zależności od:**

- Ceny
- geolokalizacja centrum danych
- Wirtualna ilość rdzeni procesora (1-8)
- Wirtualna ilość pamięci RAM (1-32GB)
- Wielkość dysku SSD (20GB-1TB)

**Istnieje kilkadziesiąt wariantów serwerów dedykowanych w zależności od:**

- Ceny
- geolokalizacja centrum danych
- Ilość rdzeni procesora (8..32)
- Ilość pamięci RAM (32GB..512GB)

- Wielkość dysku SSD (100TB-400TB)

**Platforma IoT @City jest dedykowana dla pojedynczego odbiorcy (dalej zwanego klientem):**

- lokalne władze, urzędy, gminy, powiaty
- zakłady przemysłowe
- B2B (podmioty biznesowe)

Ponieważ serwery nie są współdzielone między klientami, upraszcza to dostęp, zabezpieczenia i problemy związane wydajnością. Z tego powodu tylko jeden klient jest odpowiedzialny za efektywne bezpieczeństwo, stabilność, wydajność, przepustowość danych itp. W przypadku niewystarczającej wydajności, klient może zakupić wyższy plan (VPS lub serwer dedykowany), bardziej optymalny dla wymaganej funkcjonalności i wydajności.

W szczególnych przypadkach komunikacja „chmura z chmurą”, może zostać wdrożona do globalizacji i centralizacji danych do większych obszarów zamiast chmury wielu klientów.

### 1.5.1. Serwer @City i bramki komunikacyjne TCP/UDP

Komunikacja z serwerem @City jest realizowana w oparciu o niskopoziomową aplikację pracującą na serwerze dla maksymalizacji wydajności.

Główne cechy aplikacji @City Serwer to:

- asynchronicznie odbieranie dane z urządzeń (poprzez protokoły: UDP, TCP IP, HTTP)
- zminimalizowanie wykorzystania danych i jego kosztów między urządzeniami a chmurą (za pomocą niskopoziomowych protokołów komunikacyjnych UDP, TCP IP)
- odbieranie zaszyfrowanych statusów urządzeń (za pośrednictwem dowolnych łącz komunikacyjnych)
- autoryzacja i weryfikacja danych z urządzeń i odszyfrowanie ich statusów
- dekodowanie i aktualizowanie stanu urządzeń oraz bezpośredni zapis do tabel MariaDB/MySQL (dane nie przetworzone)
  - tabela aktualnych danych (zawiera tylko najnowszy stan każdego urządzenia)
  - tabele historycznych danych (zawiera wszystkie statusy dla jednego urządzenia)
- przesyłanie oczekujących komend (Events) do urządzeń
- zapewnienie weryfikacji i potwierdzeń

**Oprogramowanie serwera @City jest takie samo dla każdego użytkownika i nie można go dostosować do różnych klientów.**

### 1.5.2. Integracja HTTP LoRaWAN

Sterowniki LoRaWAN są zintegrowane z chmurą @City poprzez interfejs HTTP (webhooks) dostępny w serwerze sieciowym/aplikacyjnym LoRaWAN.

Obsługiwanych jest kilka rodzajów serwera sieciowego/aplikacyjnego:

- TTN (ograniczony czas “On The Air” oraz maksymalna ilość komend przesłanych do sterownika)
- LoRaWAN-Stack (Wymaga posadowienia na fizycznym urządzeniu z dostępem do internetu).



- LoraServer.Io (tylko przesyłanie danych do servera)

Chmura @City dla sterowników LoRaWAN podzielona analogicznie jak dla innych interfejsów I jest omówiona w poprzednim rozdziale.

### 1.5.3. Interfejs Frond-end

Interfejs pomiędzy bazą danych a aplikacją front-end jest realizowany za pomocą skryptów PHP, do wyodrębniania niestandardowych danych z bazy danych Chmury @City. Wykorzystuje on bardzo elastyczny mechanizm wyszukiwania, oparty na oryginalnych zapytaniach SQL, aby ograniczyć ilość danych do dalszej obróbki. Interfejs dostarcza wyniki zapytań w formacie JSON w celu dalszego dekodowania i przetwarzania przez aplikację web'ową interfejsu użytkownika w JavaScript.

**Oryginalny interfejs front-end jest taki sam dla każdego użytkownika i nie można go dostosować do różnych klientów.**

Interfejs nakładki może zostać utworzony przez naszych pracowników lub we współpracy, aby zapewnić dostosowanie systemu do potrzeb klienta.

### 1.5.3. Uprawnienia Serwera @City

Fizyczne uprawnienia klienta (do Serwera) są ograniczone.

**Dostęp do plików tylko dla katalogu 'templates' (rodzime pliki tekstowe - .txt, .js, .css, .html):**

- dodawanie, usuwanie, modyfikowanie zwykłych plików HTML (Front-End GUI / Adaptacja interfejsu użytkownika)
- dodawanie, usuwanie, modyfikowanie zwykłych plików JavaScript (Front-End GUI / Rozwój interfejsu i algorytmy)
- dodawanie, usuwanie, modyfikowanie zwykłych plików CSS (Front-End – Kastomizowane widoki / motywy)
- dodawanie, usuwanie, modyfikowanie plików tekstowych szablonów (Front-End GUI)
- dodawanie, usuwanie, modyfikowanie zakładek, skrótów, linków do wyników danych

**Inne prawa dostępu:**

- Pełny dostęp do bazy danych chmury @City (MySQL/MariaDB), gdzie przechowywane są wszystkie dane urządzeń
- Dostęp do usług internetowych jest określony poprzez konto użytkownika (dane logowania) dla klienta.
- Klient nie może przekazywać danych logowania osobom trzecim (wielokrotny dostęp do wyników może wpłynąć na wydajność, stabilność i bezpieczeństwo całego systemu)
- W szczególnych okolicznościach, przy korzystaniu z wysokowydajnego serwera dedykowanego, można skonfigurować ogólnodostępne konto publiczne, aby publicznie udostępnić aktualne dane (nie historyczne).
- Użytkownik może powielić dane na swoim własnym serwerze MySQL i przeprowadzić własną analizę i przetwarzanie danych, aby nie wpływać na wydajność serwera @City

**Personel iSys - Intelligent Systems - ma nieograniczony dostęp do całego serwera, w tym konta**



roota i pełnego dostępu do bazy danych w celu konserwacji.

W pewnych okolicznościach iSys może przyznać dodatkowe ograniczone prawa do klienta (skrypty PHP, pliki), po sprawdzeniu kodu źródłowego, uruchomieniu testów, jeśli nie wpłynie to na ogólne bezpieczeństwo systemu, stabilność i wydajność.

## **1.6. Inteligentne urządzenia IoT, CloT**

### **1.6.1. CloT – urządzenia GSM**

Nasze urządzenia zawierają mikrokontroler i moduł GSM/GPS/GNSS (2G..4G, NBIoT, CATM1) do komunikacji. Mikrokontroler zawiera zaszyfrowany bootloader do bezpiecznej aktualizacji oprogramowania OTA. Umożliwia to tworzenie wielu wariantów systemu opartych na tym samym „urządzeniu inteligentnym CloT”. Urządzenia wymagają instalacji karty SIM na wybranego operatora oraz wymagany zakres usług. Alternatywnie może zostać wmontowany MIM (sim elektroniczny) na etapie produkcji urządzenia. Na każdą kartę SIM/MIM nałożone są opłaty abonamentowe operatora.

### **1.6.2. IoT – urządzenia LoRaWAN**

LoRaWAN umożliwia transmisję danych na bardzo duże odległości (do ok 15km). Zasięg ten zależy od prędkości przesyłu danych, ilości danych, urbanizacji terenu i sprawności torów radiowych urządzeń. Nasze urządzenia zawierają mikrokontroler i moduł **LoRaWAN** do komunikacji. Mikrokontroler zawiera zaszyfrowany bootloader do bezpiecznej aktualizacji oprogramowania OTA. Umożliwia to tworzenie wielu wariantów systemu opartych na tym samym „urządzeniu inteligentnym IoT”. Urządzenia pracują w paśmie otwartym ISM bez dodatkowych opłat abonamentowych. Konieczne jest stosowanie Bramek LoRaWAN (Gateway) zapewniających pokrycie całego obszaru z dostępem do internetu. W przypadku istniejących bramek LoRaWAN w zasięgu urządzeń (skonfigurowanych na server TTN), możliwe jest przesyłanie informacji za ich pośrednictwem.

### **1.6.2. BAS, BMS, IoT – urządzenia Ethernet, WiFi**

Sterowniki Ethernet i WiFi umożliwiają komunikację z systemem @City opartą na IP (bez pobierania opłat za transmisję danych od operatora GSM). Urządzenia wykorzystują istniejące łącza lokalne LAN/WLAN/WAN z dostępem do internetu.

Urządzenia te mają również zaszyfrowany bootloader, a urządzenia mogą być aktualizowane poprzez główny interfejs komunikacyjny. W przypadku WiFi uaktualnienie firmwaru przez OTA jest z głównego serwera WWW.

## **1.7. Warianty współpracy Business to Business (B2B)**

Istnieje kilka opcji współpracy biznesowej dotyczącej urządzeń:

- import / eksport gotowych produktów (rozdzielnie, gotowe urządzenia) - produkty końcowe
- import / eksport produktów OEM (PCB, sterowniki itp.) - produkty pośrednie, części zamienne

- franczyza - produkcja na naszej licencji dla rynków lokalnych (dostarczamy tylko zaprogramowane mikro-kontrolery) I sprzedaż pod znakami handlowymi iSys – Intelligent Systems.
- Koprodukcja - (jak powyżej), ale sprzedaż pod zewnętrznymi markami urządzenia są sprzedawane jednorazowo, stają się własnością klienta.

**Urządzenia nie podlegają opłatom abonamentowym.**

Dla każdego klienta konieczne jest wykupienie mini-chmury @City na każdy rok użytkowania systemu (hosting VPS lub serwer dedykowany – zwany dalej **hostingiem**). Konieczne jest coroczne I terminowe opłacanie hostingu, gdyż posiada on stały I dedykowany adres IP. Urządzenia łączą się z serwerem przy pomocy tego adresu IP. Adres jest utrzymywany tylko przez okres trwania usługi. Wyłączenie usług skutkuje utratą tego adresu IP I wymaga demontażu, przekonfigurowania I powtórnego montażu wszystkich urządzeń.

W przypadku wyboru serwera należy się zdecydować na typ serwera (VPS lub dedykowany).

W przypadku zbyt małej wydajności można w każdej chwili zwiększyć plan hostingowy (RAM, SDD, ilość procesorów). Jest to zmiana jednokierunkowa I nie ma możliwości zmniejszenia planu hostingowego.

## 2. Funkcjonalność platformy IoT @City

Platforma @City obsługuje konfigurowalny szablon Front-End do wizualizacji danych, zapytań, ograniczania i przetwarzania (dane bieżące / historyczne):

- wizualizacja wybranych danych na mapie (geolokalizacja)
- wyświetlanie wybranych danych i wyników w tabelach
- wizualizacja wybranych danych na wykresach słupkowych
- wizualizacja wybranych danych na wykresach historycznych
- umożliwia eksport danych do aplikacji innych firm i dalsze przetwarzanie
- inne funkcje będą aktualizowane na indywidualne potrzeby.

Interfejs użytkownika jest dostępny za pośrednictwem statycznego adresu IP. Dodatkowo mogą być utworzone domeny/subdomeny z przekierowaniem DNS na ten adres, aby uprościć nazwę w przypadku platformy front-end.

Przykładowa i demonstracyjna instalacja jest dostępna (tylko dla potencjalnych klientów). Poinformuj nas, kiedy chcesz testować system - aby umożliwić publiczny dostęp do platformy. **Może wymagać statycznego adresu IP komputera zdalnego, aby umożliwić komunikację z platformą @City.**

### 3. Strona Główna

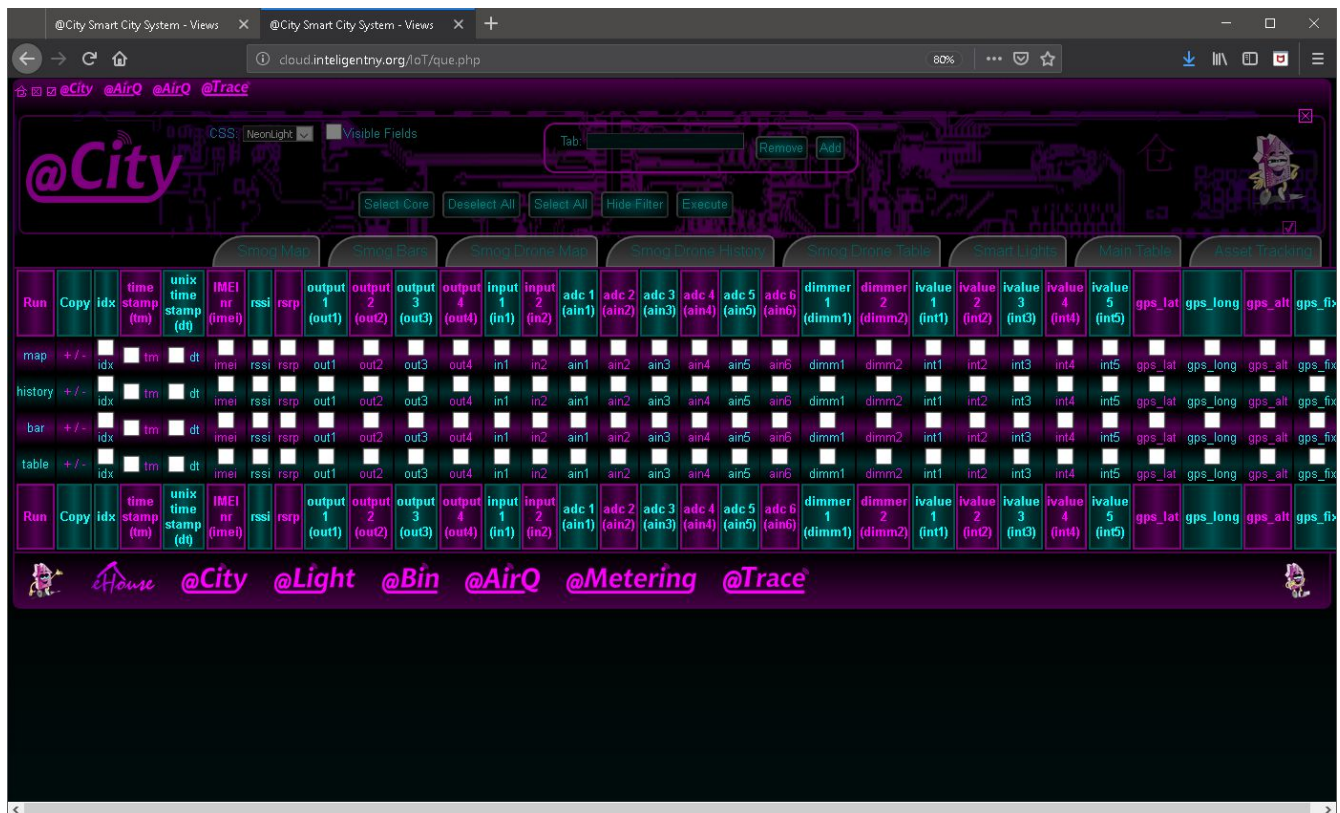
Strona Główna WWW serwisu jest celowo pusta ze względów bezpieczeństwa: <http://%YourIP%/IoT/>

Może być indywidualnie utworzona, zawierać linki do wykorzystanych funkcji I wykresów platformy @City jeśli jest taka potrzeba.

### 4. Formularz główny

Formularz główny ma na celu łatwe tworzenie nowych ustawień, wykresów, tabel, itd:

<http://iot.inteligentny.org/IoT/que.php>



Opisy (Od góry I od lewej do prawej)

## **4.1. Nagłówek**

**4.1.1. Home Link - (otwiera aktualną tabelę)**

**4.1.2. 'X' pole – otwiera/zamyka formularz wyszukiwania I opcji**

**4.1.3. 'V' pole – otwiera/zamyka Formularz filtra pól**

**4.1.4. Ikony - linki do rezultatów wizualizcji (edytowalne)**

## **4.2. Formularz:**

**4.2.1. “X” pole - otwiera/zamyka cały formularz wyszukiwania I opcji**

**4.2.2. “CSS” – Wybór motywu wizualizacji**

Modyfikuj motyw wizualizacji Plik CSS musi istnieć w katalogu „templates/css/” - automatycznie wyświetlany na liście.

**4.2.3. “Visible Fields” – pokazuje/ukrywa formularz filtra pól**

**4.2.4. “Tab” - Nazwa zakładki do dodawania I usuwania**

**4.2.5. “Add / Remove” przyciski – Dodaje I usuwa zakładki z nazwą znajdującą się w polu Tab**

**4.2.6. “Select Core” - główne pola**

Wybierz główne pola widoczne w tabeli. Widok jest aktualizowany automatycznie.

**4.2.7. “Deselect All” - odznacz wszystkie pola**

Odznacz wszystkie pola (następnie należy je wybrać ręcznie)

**4.2.7. “Select All” - wybierz wszystkie pola**

Zaznacz wszystkie pola (należy po nich ręcznie odznaczyć niektóre)

**4.2.8. “Hide Filter” – Ukryj formularz**

Jest to odpowiednik pola (X)

#### 4.2.9. “Execute” – uruchomienie zmiany parametrów

#### 4.2.10. 'V' pole – pokazuje/ukrywa Filtr Pól.

### **4.3. Zakładki**

Indywidualnie tworzone zakładki z nazwami i ustawieniami wstępnymi (przechowywane w pliku “`cfg/tabs.cfg`”).

Plik zawiera nazwę i adres URL (oddzielone znakiem tabulatora).

### **4.4. Zawartość tabeli**

Wyświetla wszystkie pola ograniczone przez filtr pól.

Pola w tabeli:

#### 4.4.1. Run – typ widoku rezultatów

**map** – wyniki rozmieszczane na mapie (można wybrać jedno lub więcej pól)

**history** - wykresy historyczne (można wybrać jedno lub więcej pól)

**tab** - wyświetla tabelę (można wybrać dowolną kombinację pól)

**bar** - tylko jedno pole jest wyświetlane na wykresie słupkowym

Po naciśnięciu jednej pozycji z rzędu otworzy nowe wyniki z wybranymi polami (dla bieżącego wiersza).

#### 4.4.2. Copy (+/- linki)

Dodawanie / usuwanie zakładki o nazwie ustawionej w polu Tab. Używa tylko pól zaznaczonych w tym samym wierszu tabeli.

#### 4.4.3. Linki w komórkach tabeli

Naciśnięcie dowolnej innej nazwy pola zainicjuje wizualizację danych bieżącego pola dla wybranego wiersza (typu wykresu).

### ***4.5. Kolejność danych***

Kolejność wyświetlanych pól jest zgodna z kolejnością w polach tabeli (jednak pole `tm` jest zawsze wysyłane na koniec tekstu). Kolejność można zmienić tylko przy bezpośredniej edycji parametrów URL wywołania strony.

#### 4.6. Przykład

Na przykład: ustawienie zakładki z nazwą „Śledzenie zasobów”, które ma zawierać mapę z datą/czasem i prędkością na mapie

Cały opis odnoszący się do wiersza, w którym tekst „Map” znajduje się w kolumnie „Uruchom”.

- 1) Wprowadź nazwę „Śledzenie zasobów” w polu Tab (bez znaków cudzysłowu)
- 2) Upewnij się, że wszystkie kolumny są odznaczane w wierszu
- 3) wybierz tylko tm, gps\_speed\_km w wierszu
- 4) naciśnij przycisk '+' w wierszu



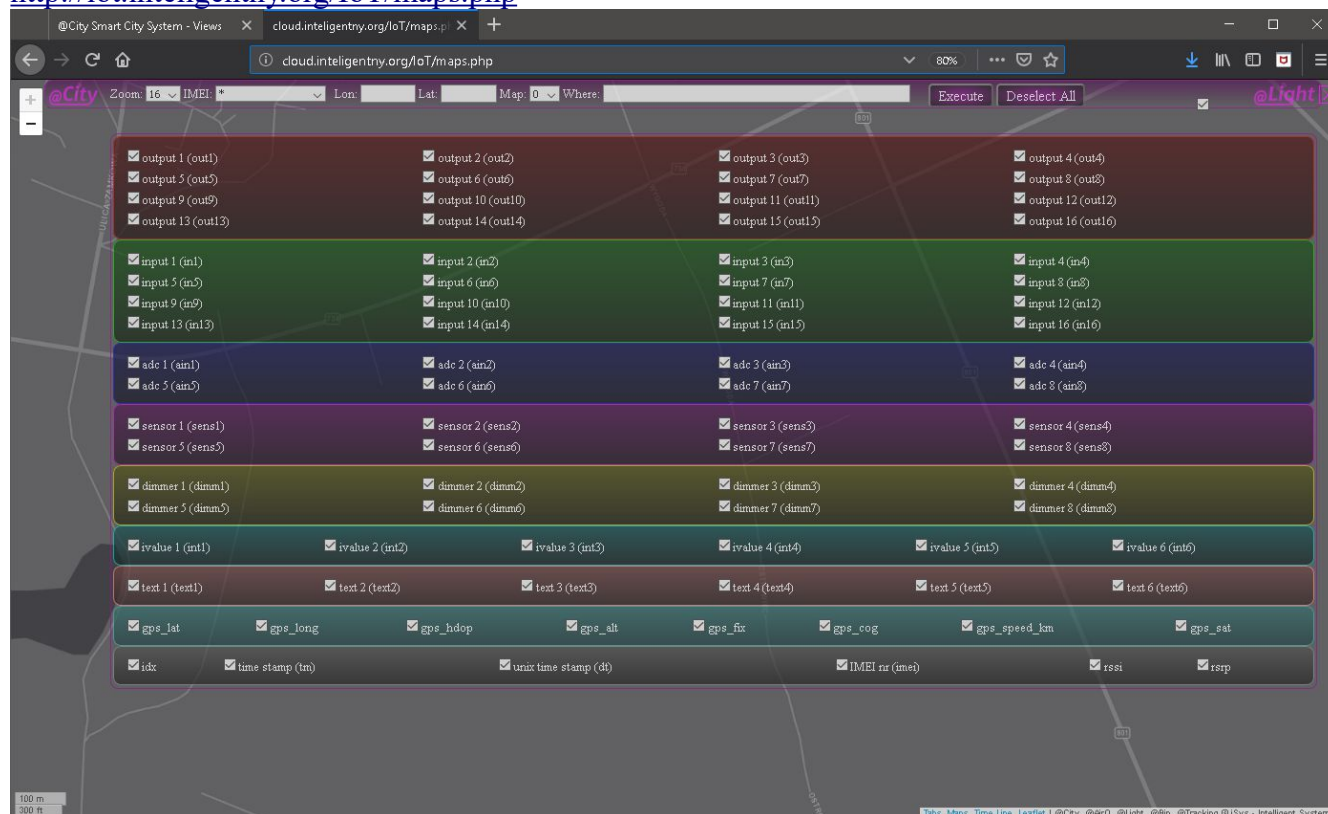
## 5. Mapy

Mapy można uruchamiać z Formularza głównego z konfiguracją wstępną

### 5.1. Inicjalizacja Mapy

Inicjalizacja mapy jest wykonywana ręcznie, gdy jest mapa uruchomiona bezpośrednio za pomocą linku:

<http://iot.inteligeny.org/IoT/maps.php>



1. Użytkownik powinien odznaczyć wszystkie pola (naciśnij przycisk Odznacz “Deselect All”)
2. Naciśnij pole wyboru dla wyświetlanych pól (np. Ain5 (dla poziomu Smogu) i tm (dla daty/godziny pomiaru))
3. Naciśnij pole wyboru „V”, aby ukryć formularz pól
4. Naciśnij przycisk Execute, aby uruchomić zapytanie DB i wyświetlić bieżące informacje ze wszystkich czujników/urządzeń
5. Mapa z danymi jest aktualizowana po 30 sekundach lub dłużej.

### 5.2. Dodatkowe ustawienia zapytań

Ustawienia opisane od lewej do prawej (na powyższym obrazku).

### 5.2.1. Zmiana skali Mapy (Zoom Level)

1. Poziom powiększenia można modyfikować za pomocą przycisków (+/-) dla skali ( $\text{aktualna\_skala} \times 2$  lub  $\text{aktualna\_skala} / 2$ ). Naciśnięcie jednego z tych przycisków automatycznie zmodyfikuje skalę.
2. Innym sposobem jest wybranie wartości powiększenia w polu wybieralnym “Zoom” i naciśnięcie przycisku Execute. W tym przypadku cały widok/mapa jest przeładowywany i odświeżany (zajmuje trochę czasu podczas inicjalizacji).

### 5.2.2. IMEI (Wybór urządzenia)

Pole IMEI zawiera unikalny identyfikator urządzenia lub unikalny alias dla urządzenia. Domyślne ustawienie to \* (gwiazdka), która pokazuje najnowsze wartości i geolokalizację dla każdego urządzenia.

Ustawienie dowolnej wartości IMEI spowoduje wyświetlenie danych historycznych wybranego urządzenia. Ma to sens tylko w przypadku czujników mobilnych i zmieniających pozycję. W przeciwnym razie wyniki będą nakładać się na mapę w tej samej pozycji.

### 5.2.3. Lon, Lat (Długość, szerokość geograficzna - współrzędne)

Ustawia środkową pozycję mapy. To pole przechwytuje pozycję kursora po naciśnięciu przycisku myszy na mapie.

### 5.2.4. Modyfikacja Stylu Mapy (Motywu)

Styl/motyw mapy można wybrać z pola wyboru **Map** (np. Ciemny, Szary, Topograficzny). Różne motywy map mogą mieć różne maksymalne poziomy powiększenia, więc może wymusić odpowiedni motyw, aby zwiększyć skalę mapy.

### 5.2.5. Klauzula WHERE

Klauzula “WHERE” jest używana do dodatkowego ograniczenia zapytania {część WHERE} dla MySQL/MariaDB.

Ta klauzula jest brana pod uwagę przy konstruowaniu pełnego ciągu zapytania (QUERY) dla wyniku z bazy danych. Może ograniczać dane, czas i inne wartości, zmniejszając liczbę wyników. W tym polu muszą być używane oryginalne nazwy pól tabeli (nie alias). Na przykład:

1. **gps\_speed\_km>10** //Prędkość jest większa niż 10 km/h
2. **ain5>3** // ain5 jest większe niż 3 (zawiera liczby cząstek stałych 2.5um - poziom smogu)
3. **gps\_speed\_km>10 and ain6>5** //Prędkość jest większa niż 10 km/h, ain6 jest większa niż 5 (trzymaj 10um cząstek - poziom smogu)

### 5.2.6. Execute (Uruchamia zapytanie)

Naciśnięcie tego przycisku jest wymagane, aby zmienić dowolne ustawienia, parametry (z wyjątkiem naciśnięcia przycisków +/-).

Mapa jest ładowana od początku z nowymi ustawieniami.

Mapa nie jest w ogóle ładowana, gdy nie są dostępne żadne dane dla bieżącego zapytania.

### 5.2.7. Deselect All (Wyłącza wszystkie pola filtra)

Po naciśnięciu tego przycisku należy wybrać co najmniej jedno pole, aby wyświetlić wyniki na mapie.

### 5.2.8. 'V' (Otwórz/zamknij Formularz Filtra)

To pole wyboru służy do pokazywania / ukrywania wyboru pól do wyświetlenia.

### 5.2.9. 'X' (Pokaż/ukryj formularz zapytania i opcji)

To pole wyboru umożliwia ukrycie całego formularza wyszukiwania

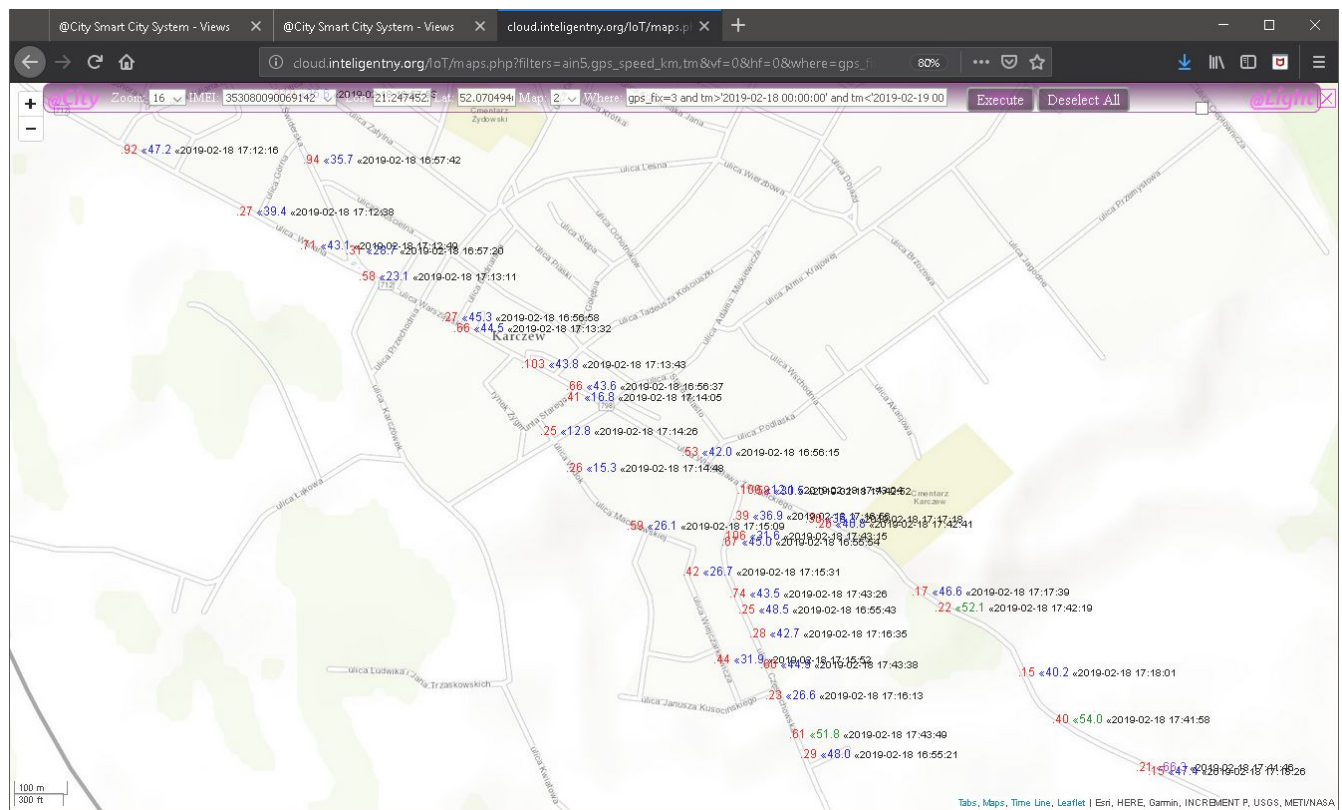
Wyniki na mapie są automatycznie odświeżane i aktualizowane o nowe wartości.

## 5.3. Przykład

Np. Przedstawienie wyników smogu na mapie (czujnik zainstalowany w samochodzie): poziom smogu – cząstek stałych 2,5um (Ain5), prędkość (gps\_speed\_km), data/godzina (tm), mapa (2 - topografia), poziom powiększenia 16.

Klauzula "WHERE" ma postać:

"gps\_fix=3 and tm>'2019-02-18 00:00:00' and tm<'2019-02-19 00:00:00' and gps\_speed\_km>0".  
//GPS= prawidłowe wyniki 3D & data=2019-02-18 & prędkość > 0 km/h



## 6. Wizualizacja wyników w tabeli

Pokazuje wyniki w tabeli.

W „Formularzu głównym” naciśnij pozycję „Visible Fields”, po wybraniu niektórych pól do wyświetlenia wstępnie skonfigurowanej tabeli

The screenshot shows the @City IoT Cloud Platform interface. At the top, there's a navigation bar with tabs for @City, @AirQ, @Trace, and @Light. Below this, there's a search bar and a table of data. The table has columns for idx, time\_stamp (tm), IMEI nr (imei), rssi, rsrp, adc 5 (ain5), adc 6 (ain6), ivalue 1 (int1), ivalue 2 (int2), gps\_lat, gps\_long, gps\_alt, gps\_fix, gps\_speed\_km, and gps\_sat. The data is sorted by idx in descending order. Below the table, there's a navigation bar with tabs for @House, @City, @Light, @Bin, @AirQ, @Metering, and @Trace.

idx	time_stamp (tm)	IMEI nr (imei)	rssi	rsrp	adc 5 (ain5)	adc 6 (ain6)	ivalue 1 (int1)	ivalue 2 (int2)	gps_lat	gps_long	gps_alt	gps_fix	gps_speed_km	gps_sat
1	2019-02-10 12:56:23	351580051067110	91	99	0	44	-16776767	450	0000.0000N	00000.0000E	4	3	0.0	07
3	2019-03-12 21:10:17	353080090069142	72	1	45	93	19	115	5202.7354N	02115.8172E	86.6	3	0.0	10
6	2019-02-20 09:13:04	356345080018095	88	102	115	73	-16777186	60	5202.9500N	02115.8000E	86.5	3	0.0	10
8	2019-03-08 10:48:03	356345080006819	120	109	4	5	98736	5391744	5202.7364N	02115.8197E	88.2	3	0.0	05
11	2019-03-07 13:09:22	karczew	76	76	103	62	19	77	5203.9312N	02115.4106E	88.2	3	0.0	09

### 6.1. Inicjalizacja tabeli

Jeśli tabela jest otwierana bezpośrednio z linku <http://iot.intelgentny.org/IoT/que.php?func=tabs> wymaga wcześniejszej inicjalizacji ustawień.

Możesz zaznaczyć pola, które mają być widoczne (naciskając „Visible Fields”).



1. Wybierz wszystkie wymagane pola dla wyświetlanych pól
2. Naciśnij pole wyboru „Visible Fields”, aby ukryć formularz filtra pól
3. Naciśnij przycisk “Execute”, aby uruchomić zapytanie do bazy danych i tabelę wyświetlania

## 6.2. Opcjonalne ustawienia zapytania

Ustawienia są opisane od lewej do prawej (na rzucie ekranu).

### 6.2.1. Sort - sortowanie pola rosnąco/malejąco

Pole sortowania jest równoważne naciśnięciu nagłówka kolumny.

### 6.2.2. DB / IMEI – Wybór urządzenia/tabeli

Pole IMEI zawiera unikalny identyfikator urządzenia lub unikalny alias dla urządzenia. Z pustą wartością pokazuje tabelę najnowszych wartości.

Ustawienie dowolnej wartości IMEI spowoduje wyświetlenie danych historycznych wybranego urządzenia.

### 6.2.3. CSS – wybierz styl (motyw wizualizacji)

### 6.2.4. “Visible Fields” – Pokaż/zamknij formularz filtra pól

### 6.2.5. “Remove Empty” – Nie pokazuje pustych kolumn

### 6.2.6. 'X' pol (pokaż/zamknij formularz zapytań)

### 6.2.7. Klauzula Where (ograniczanie danych)

To jest dodatek dla dodatkowego ciągu zapytań MySQL / MariaDB {WHERE}

Ta klauzula jest brana pod uwagę przy konstruowaniu pełnego zapytania {QUERY} do ograniczenia wyniku bazy danych. Może ograniczać dane, czas i inne wartości, zmniejszając liczbę wyników. W tym polu muszą być używane oryginalne nazwy pól tabeli (nie alias). Na przykład:

1. **gps\_speed\_km>10** //Prędkość jest większa niż 10km/h
2. **ain5>3** // ain5 jest większe niż 3 (trzyma 2.5um liczby cząstek - poziom smogu)
3. **gps\_speed\_km>10 and ain6>5** //Prędkość jest większa niż 10 km/h, ain6 jest większa niż 5 (zawiera 10um cząsteczki - poziom smogu)

### 6.2.8. “Select Core” (Włącz większość popularnych pól)

### 6.2.9. “Deselect All” (Usuń wszystkie pola z zapytania)

Po naciśnięciu tego przycisku należy wybrać co najmniej jedno pole, aby wyświetlić wyniki na mapie.

### 6.2.10. “Execute” (Uruchom zapytanie)

Naciśnięcie tego przycisku jest wymagane, aby zmienić dowolne ustawienia, parametry. Tabela jest przeładowywana od początku z nowymi ustawieniami wstępnymi.

### 6.2.11. 'V' pole (otwórz/zamknij formularz filtra pól)

To pole wyboru służy do pokazywania/ukrywania pól do wyświetlenia.

Wyniki w tabeli są sortowane zgodnie z ustawieniem pola “Sort”. Kolejność sortowania można zmienić, naciskając nagłówek wiersza (raz dla jednego kierunku, kolejny raz dla przeciwnego kierunku).

Niektóre wyniki w kolumnach są linkami, prowadzą do kolejnych ekranów wizualizacyjnych



(deklarowanych na stałe).

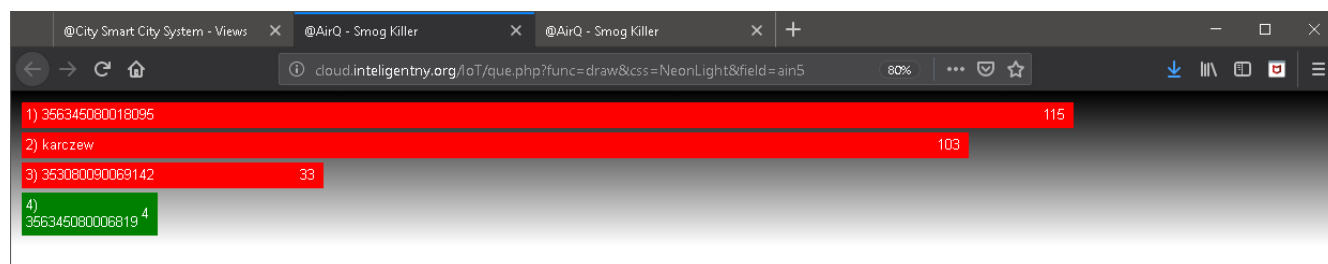
Podczas wyświetlania danych historycznych dla urządzenia należy je ograniczyć, aby nie wyświetlać całej informacji o historii, ponieważ może to prowadzić do problemów z wydajnością lub brakiem pamięci.

## 7. Wykresy Słupkowe.

Wykresy słupkowe należy uruchamiać z formularza głównego, naciskając pojedyncze pole w wierszu „Bar”.

Wyświetla posortowane słupki “znormalizowane” do maksymalnej wartości, od najwyższego do najniższego rzędu.

Jest przydatny do szybkiego sprawdzania ekstremalnych wyników i podejmowania pewnych działań.



Najeżdżenie kursorem myszy powoduje wyświetlenie dodatkowych informacji o urządzeniu

## 8. Wykresy Archiwalne.

Wykresy historyczne mogą być inicjowane z formularza głównego po naciśnięciu wybranej kolumny w wierszu „Historia” (dla pojedynczego pola).

W przypadku wyświetlania wielu pól, w wierszu „History” należy zaznaczyć żądane pola, a pole „Historia” musi zostać naciśnięte w kolumnie „Run”.

Wyniki historyczne są ograniczone do 24 godzin + 24 godziny (w przypadku ewentualnego odświeżania wykresów), gdy ręcznie nie ustanowiono żadnych limitów.

### 8.1. Inicjalizacja wykresów archiwalnych

Wykresy historyczne po otwarciu z głównego łącza wymagają inicjalizacji jak i innych wyników, gdy są otwarte bez podania parametrów .

Można wybrać wiele pól, aby wyświetlić różne elementy. Można go również ustawić w formularzu filtra pola.

The screenshot shows the @City IoT Cloud Platform interface. At the top, there's a navigation bar with the @City logo and links to @City, @AirQ, and @Trace. Below the navigation bar, there's a search bar and a filter bar. The main area contains a form for selecting fields for historical charts. The form has a table with columns for different data types: outputs, inputs, ADCs, sensors, dimmers, and I-values. Each data type has a list of fields with checkboxes next to them. The 'Visible Fields' checkbox is checked. At the bottom, there's an 'Execute' button.

1. Naciśnij wszystkie wymagane pola wyboru dla wyświetlanych pól
2. Naciśnij pole wyboru „Visible Fields”, aby ukryć formularz pól
3. Naciśnij przycisk Execute, aby uruchomić zapytanie DB i wyświetlić tabelę

## 8.2. Opcjonalne ustawienia dla wykresów historycznych

Pozycje opisane od góry i od lewej do prawej (na zrzucie ekranu).

### 8.2.1. IMEI – (Wybór urządzenia do wyświetlenia danych)

Pole IMEI zawiera unikalny identyfikator urządzenia lub unikalny alias dla urządzenia. Dla wartości “\*” (gwiazdka) pokazuje tabelę najnowszych wartości, która nie ma sensu w tym przypadku. Ustawienie dowolnej wartości IMEI spowoduje wyświetlenie danych historycznych wybranego urządzenia.

### 8.2.2. “Min” - ograniczyć minimalną wartość pierwszego pola

### 8.2.3. “Max” - ograniczyć maksymalną wartość pierwszego pola

### 8.2.4. 'V' – Pokaż/ukryj formularz filtra pól

### 8.2.5. “From” ustaw minimalną datę/godzinę daty (\*)

### 8.2.6. “To” ustaw maksymalną datę/godzinę daty (\*)

### 8.2.7. 'X' pole (Pokaż/ukryj formularz zapytań)

### 8.2.8. Klauzula “Where”

Klauzula ograniczająca wyniki wyszukiwania MySQL/MariaDB - dodatkowy ciąg zapytania SQL {WHERE }.

Ta klauzula jest brana pod uwagę przy konstruowaniu pełnego ciągu QUERY dla tabeli. Może ograniczać ilość danych: czas lub inne wartości. W tym polu musi być użyta poprawna składnia SQL oraz oryginalne nazwy pól tabeli (nie alias). Na przykład:

1. **gps\_speed\_km>10** //Prędkość jest większa niż 10 km/h
2. **ain5>3** // ain5 jest większe niż 3 (zawiera 2.5um liczby cząstek - poziom smogu)
3. **gps\_speed\_km>10 and ain6>5** //Prędkość jest większa niż 10 km/h, ain6 jest większa niż 5. Ain6 - zawiera pomiar cząstek stałych 10um (poziom smogu).

### 8.2.9. “Deselect All” (Usuń wszystkie pola z zapytania)

Po naciśnięciu tego przycisku co najmniej jedno pole musi zostać wybrane ręcznie, aby wyświetlić wyniki historyczne.

### 8.2.10. "Execute" (uruchom zapytanie)

Naciśnięcie tego przycisku jest wymagane, aby zmienić dowolne ustawienia, parametry (z wyjątkiem wyświetlania pól lub panelu zapytań). Tabela jest przeładowywana od początku z nowymi ustawieniami wstępnymi.

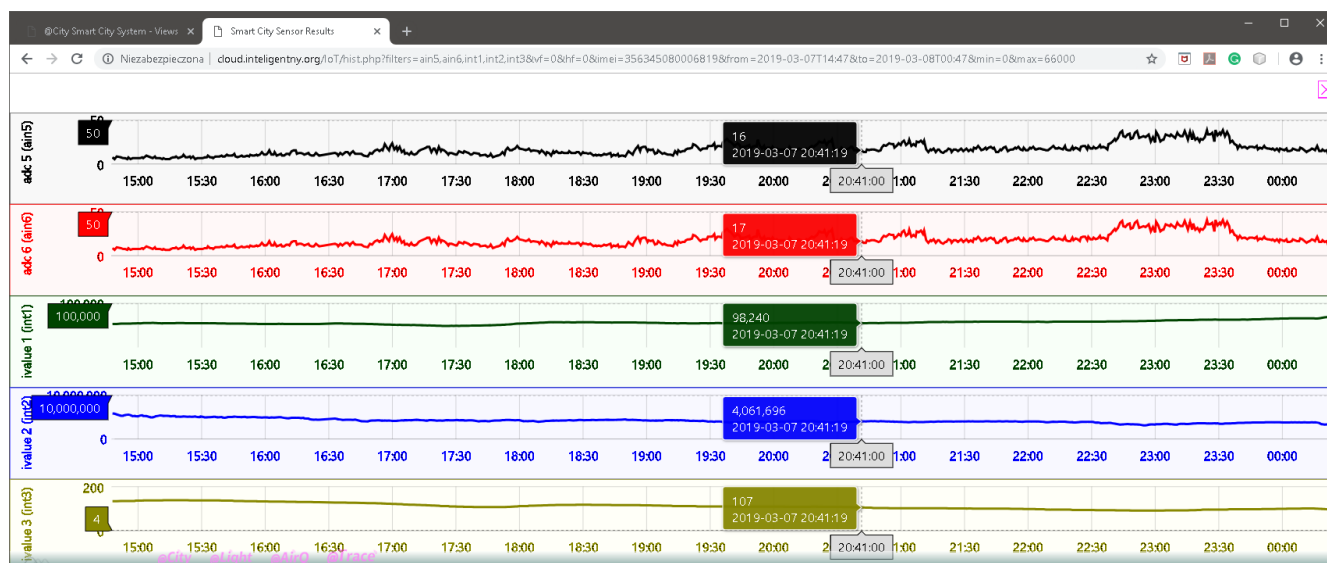
### 8.2.11. 'V' Pole wyboru (otwórz/zamknij formularz filtra pól)

To pole wyboru służy do pokazywania/ukrywania wyboru pól do wyświetlenia.

## 8.3. Wykres Słupkowy : (wyświetla tylko istniejące dane)



## 8.4. Wykres ciągły: (dla tych samych danych wejściowych):



Przesunięcie wskaźnika myszy wyświetla wartości pomiarów oraz datę/godzinę.

## 9. Zgodność z przeglądarką internetową

Funkcja / Przeglądarka WWW	Chrome 72	Fire Fox 65	Edge	O
Maps (mapy)	+	+	+	+
Historical (dane archiwalne)	+	+	+	+
Bars (Wykres słupkowy)	+	+	+	+
Tabs (Tabele)	+	+	+	+

\* - Firefox nie obsługuje wyboru daty/godziny (pole tekstowe musi być ręcznie edytowane przy użyciu odpowiedniego formatu daty i czasu).

Internet Explorer nie jest obsługiwany (zamiast tego użyj Edge)

Inne przeglądarki internetowe nie były testowane.

## 10. Kustomizacja widoku/motywu

Strony internetowe są oparte na ogólnym pliku szablonu znajdującym się w katalogu '**templates**' '\*.template'.

Dodatkowo każdy typ strony zawiera:

- 1) plik „\*.head”, który przechowuje nagłówek strony (linki, importowane CSS, pliki JavaScript itp.)
- 2) pliki „\*.foot”, które przechowują stopkę strony (linki itp.)

Motyw wizualizacji można zmienić zgodnie z preferencjami użytkownika, kopiując i modyfikując pliki CSS. Pliki CSS znajdują się w katalogu '**templates/css**'. Różne motywy stron internetowych mogą być używane do tworzenia zoptymalizowanych szablonów dla np. drukowania, smartfonów, PAD.

**Widoki tabeli** - mają pole do wyboru do wyboru pliku CSS do pełnej modyfikacji motywu (przechowywane w katalogu '**templates/css/tabs**').

The screenshot shows the @City Smart City System interface. At the top, there are tabs for different views: @City, @AirQ, @AirQ, and @Trace. Below the tabs, there are filters for Sort (tm), DB, and CSS (zEmpty). There are also checkboxes for Visible Fields and Remove Empty. A 'Where (SQL):' field is present. Below the filters, there are buttons for Execute, Hide Filter, Select All, Deselect All, and Select Core. The main part of the interface displays a table with the following columns: idx, time stamp (tm), IMEI nr (imei), rssi, rsrp, dimmer 1 (dimn1), dimmer 2 (dimn2), dimmer 3 (dimn3), ivalue 1 (int1), ivalue 2 (int2), gps\_lat, gps\_long, gps\_alt, gps\_fix, gps\_speed\_kn, and gps\_sat. The table contains several rows of data, including timestamps, IMEI numbers, and various sensor readings. At the bottom of the table, there are links for @House, @City, @Light, @Bin, @AirQ, @Metering, and @Trace.

idx	time stamp (tm)	IMEI nr (imei)	rssi	rsrp	dimmer 1 (dimn1)	dimmer 2 (dimn2)	dimmer 3 (dimn3)	ivalue 1 (int1)	ivalue 2 (int2)	gps_lat	gps_long	gps_alt	gps_fix	gps_speed_kn	gps_sat
3	2019-03-13 07:20:55	353080090069142	78	1	0	255	255	19	115	5202.7326N	02115.8134E	87.5	3	0.0	07
8	2019-03-08 10:48:03	356345080006819	120	109	0	134	0	98736	5391744	5202.7364N	02115.8197E	88.2	3	0.0	05
11	2019-03-07 13:08:22	karczew	76	76	0	255	255	19	77	5203.9312N	02115.4106E	88.2	3	0.0	09
6	2019-02-20 09:13:04	356345080018095	88	102	0	255	255	-16777186	60	5202.9500N	02115.8000E	86.5	3	0.0	10
1	2019-02-10 12:56:23	351580051067110	91	99	255	255	255	-16776767	450	0000.0000N	00000.0000E	4			

**Widoki mapy** - ogólny motyw jest wybierany przez pole „map”. Dodatkowo istnieje domyślny plik CSS '**templates/css/map.css**', który zawiera pewne dodatkowe funkcje, takie jak ukrywanie/kolorowanie wyników na podstawie jego wartości. Reszta tego pliku CSS jest praktycznie ograniczona do formularzy zapytań i pól.

Większość plików PHP platformy @City do wizualizacji akceptuje parametr **css** z wartością nazwy pliku dla motywu (bez rozszerzenia). Plik musi znajdować się w katalogu '**templates/css**', a w nazwie rozróżniana jest wielkość liter.

Niektóre elementy wyświetlania motywu znajdują się bezpośrednio w pliku JavaScript znajdującym się w katalogu „**template/js**”.

Główny skrypt Front-End platformy @City „**@City.js**” znajduje się w górnym katalogu. W tej lokalizacji nie ma możliwości modyfikacji, jednak skrypt może zostać skopiowany do katalogu 'templates/js' i tam zmodyfikowany. Użycie indywidualnego skryptu wymaga aktualizacji wszystkich plików nagłówkowych.

## 11. Tworzenie indywidualnych algorytmów przetwarzania danych

Niektóre unikalne czujniki mogą wymagać specjalnych funkcji obliczeniowych.

Nie ma możliwości aktualizacji i utrzymania wielu wariantów oprogramowania serwera @City, interfejsu PHP Front-end, co spowodowałoby wiele problemów, wersji, błędów.

Najlepszym i najłatwiejszym sposobem osiągnięcia tego celu jest aktualizacja plików „nakładki” JavaScript w celu prawidłowego wyświetlania wartości/opisu.

Oryginalne skrypty JS to otwarty plik tekstowy, który może być dostosowany do potrzeb klienta. Jak stwierdzono w poprzednim rozdziale, należy je skopiować do katalogu '**templates/js**', w którym klient ma prawa dostępu do modyfikacji.

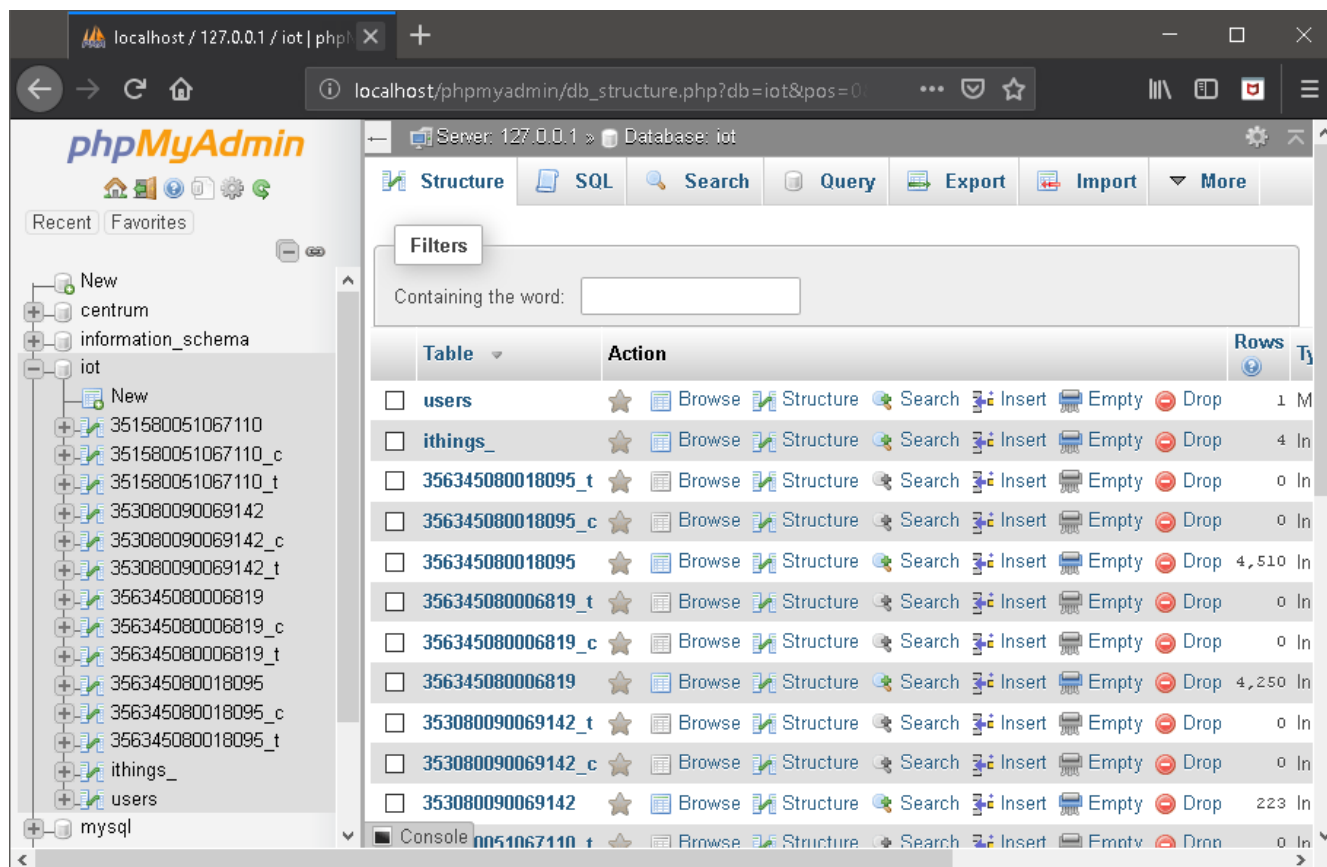
Aspekt techniczny dotyczący programowania systemu Front-End @City nie jest przedmiotem tego dokumentu, jednak programista internetowy z podstawową znajomością języka HTML i JS może dostosować aplikację Web Front-end do indywidualnych potrzeb klienta.



## 12. Struktura Bazy Danych

Baza Danych Platformy @City z nazwą 'IoT' lub '\*IoT' jest podzielona na tabele (gdzie gwiazdka jest przedrostkiem w zależności od serwera hostingowego - jeśli jest wymagany). Baza danych może być obserwowana w PHPMyAdmin (aplikacja internetowa) pod linkiem

<http://iot.inteligentny.org/phpmyadmin>



Zestaw tabel dla każdego urządzenia (gdzie \* {gwiazdka} to adres IMEI - unikalny identyfikator):

- „\*\_t” - tokeny do bieżącej autoryzacji użytkowników (nie używane w chmurach dla pojedynczego klienta)
- „\*\_c” - kolejka poleceń (zdarzeń) - do wysłania do urządzenia
- „\*\_” - wszystkie wyniki zdekodowanego statusu (archiwalne).

### Inne tabele:

Tabela „ithings\_” - zawiera zdekodowany bieżący stan wszystkich urządzeń (jest kopiowany podczas aktualizacji dowolnej z tabel \*). Struktura tabeli „ithings\_” jest prawie taka sama jak w tabelach „\*\_”. „ithings\_” ma dodatkowe pola na Adres pocztowy i opisy.

Tabela użytkowników nie może być edytowana.

### 12.1. 'Struktura tabel ithings\_' i '\*'

- `idx` - index
- `tm` - timestamp
- `dt` - unix timestamp
- `imei` - unikalny adres
- `rssi` - RSSI poziom sygnału
- `rsrp` - RSRP poziom
- `events` - comendy do przesłania do urządzenia po jego skontaktowaniu się z serwerem
- `out1` .. `out16` - dekodowane statusy wyjść
- `in1` .. `in16` - dekodowane statusy wejść
- `ain1` .. `ain8` - dekodowane statusy ADC (RAW)
- `sens1` .. `sens8` - przetworzone dane sensorów ADC (zależą od typu urządzenia)
- `dimm1` .. `dimm8` - dekodowane dane ściemniaczy
- `int1` .. `int6` - dekodowane dane liczników (zależą od typu urządzenia)
- `text1` .. `text6` - dekodowane dane (zależą od typu urządzenia)
- `creation` - data/czas utworzenia
- `last` - ostatnia aktualizacja
- `user` - future use
- `pass` - future use
- `sn` - gsm serial nr
- `status` - bieżący status
- `hash\_code` - future use
- `addr` - adres krótki (unikalny)
- `fwnr` - firmware nr
- `disabled` - disabled device (not performed)
- `gsm\_nr` - CIoT gsm number
- `vendor` - vendor code (for franchise/co-production)
- `timezone` - Time Zone offset
- `dst` - Use Daily savings settings
- `gps\_lat` - GPS Szerokość
- `gps\_long` - GPS Długość
- `gps\_hdop` - GPS HDOP
- `gps\_alt` - GPS Wysokość
- `gps\_fix` - GPS Fixation (musi być 3 – dla właściwych pomiarów 3D)
- `gps\_cog` - GPS Cog
- `gps\_speed\_km` - GPS prędkość [km/h]
- `gps\_sat` - GPS liczba satelit
- `continent`, `country`, `region`, `subregion`, `subsubregion`, `city`, `district`, `street`, `street\_nr`, `item\_nr` - Dane opisowe (!!!! Niedostępne w '\*' tabelach historycznych !!!!)
- `log` - logowanie danych

Nazwy pól są ważne przy tworzeniu zapytań SQL w celu zwiększenia funkcjonalności wyszukiwania, ponieważ oryginalna nazwa musi być używana w instrukcji SQL.

## 12.2. Kolejka komend sterujących dla urządzeń (Events) '\*\_c'

Ta tabela jest kolejką zdarzeń/poleceń dla każdego urządzenia i ma następującą strukturę:

- `command` - polecenie do wysłania lub już przesłane do urządzenia
- `potwierdzone` - flaga potwierdzenia, jeśli została już wysłana i potwierdzona
- `date` - uniksowy znacznik czasu zdarzenia
- `updated` - flaga automatycznej aktualizacji (data/czas timestamp)

## 12.3. Uzyskiwanie dostępu do wyników z baz danych - Mid-Level (odczyt danych)

Dane mogą być dostępne bez aplikacji WWW Front-end. System @City zawiera skrypt z funkcjami średniego poziomu. Wyniki są zwracane w formacie JSON.

### 12.3.1. Odczyt aktualnych statusów wszystkich sterowników

<http://%IP%/IoT/que.php?func=devsjson>

Zapytanie zwraca całą tabelę „\_ithings” (aktualne stany wszystkich urządzeń) w formacie JSON:

```
[ { "country": "", "city": "", "continent": "", "country": "", "region": "",
"subregion": "", "subsubregion": "", "city": "", "district": "", "street": "",
"street_nr": "", "item_nr": "", "gps_lat": "0000.0000N", "gps_long": "00000.0000E",
"tm": "2019-02-10 12:56:23", "creation": "2019-02-09 18:12:38", "last": "0000-00-00
00:00:00", "events": "", "user": "", "pass": "", "imei": "351580051067110",
"sn": "",
"status": "730002000000f3600330268002400000002c002c002dffffffffffffffff5b63000001c1000
001c2000000000000000000009250a4f0a760a7a0a750a780a7e00000031d032205fc34029b025c02560046
0eb305320000", "hash_code": "", "addr": "", "fwnr": "", "disabled": "",
"gsm_nr": "", "vendor": "", "timezone": "", "dst": "", "rssi": "91", "rsrp": "99",
"gps_lat": "0000.0000N", "gps_long": "00000.0000E", "gps_hdop": "", "gps_alt": "",
"gps_fix": "4", "gps_cog": "", "gps_speed_km": "", "gps_sat": "", "events": "",
"out1": "0", "out2": "0", "out3": "0", "out4": "0", "out5": "0", "out6": "0",
"out7": "0", "out8": "0", "out9": "0", "out10": "1", "out11": "0", "out12": "0",
"out13": "0", "out14": "0", "out15": "0", "out16": "0", "in1": "0", "in2": "0",
"in3": "0", "in4": "0", "in5": "0", "in6": "0", "in7": "0", "in8": "0", "in9": "0",
"in10": "0", "in11": "0", "in12": "0", "in13": "0", "in14": "0", "in15": "0",
"in16": "0", "ain1": "3894", "ain2": "51", "ain3": "616", "ain4": "36", "ain5": "0",
"ain6": "44", "ain7": "44", "ain8": "45", "sens1": "0", "sens2": "0", "sens3": "0",
"sens4": "0", "sens5": "0", "sens6": "0", "sens7": "0", "sens8": "0",
"dimm1": "255", "dimm2": "255", "dimm3": "255", "dimm4": "255", "dimm5": "255",
"dimm6": "255", "dimm7": "255", "dimm8": "255", "int1": "-16776767", "int2": "450",
```

```
"int3":"","int4":"","int5":"","int6":"0","text1":"","text2":"","text3":"","text4":"","text5":"","text6":"" } ]
```

### 12.3.2. Odczyt danych archiwalnych z urządzenia

Zapytanie o dane historyczne pojedynczego urządzenia przez numer IMEI:

<http://%IP%/IoT/que.php?func=imeijson&imei=356345080018095>

Ponieważ cała tabela może zawierać miliony wierszy, powinna być ograniczona klauzulą WHERE, aby nie blokować serwera.

Parametry dodatkowe - URL:

**func** - imeijson

**imei** - IMEI urządzenia

**field** - pola do wyświetlenia w wynikach (lista rozdzielana przecinkami)

**min** - minimalna wartość pierwszego pola z listy

**max** - maksymalna wartość pierwszego pola z listy

**sort** - pole do sortowania

**tm** - pole jest automatycznie dodawane do wyników.

**WHERE** - gdzie klauzula do ograniczenia danych

#### Przykład:

Chcemy uzyskać wyniki:

dla urządzenia z imei = **356345080018095**

pokazując pola: **ain5, ain6, gps\_lat, gps\_long**

przy limicie **ain5** w zakresie **(1, 10000)** - musi być pierwszym polem na liście

a GPS ma poprawne dane 3D (**gps\_fix = 3**)

oraz data/godzina (tm) od **2019-02-14 23:00:19** do **2019-02-15 00:00:00**

Skonstruowany ciąg URL:

[http://iot.inteligentny.org/IoT/que.php?](http://iot.inteligentny.org/IoT/que.php?func=imeijson&imei=356345080018095&field=ain5,ain6,gps_lat,gps_long&min=1&max=1000&where=gps_fix=3 and tm>'2019-02-14 23:00:19' and tm<'2019-02-15 00:00:00')

[func=imeijson&imei=356345080018095&field=ain5,ain6,gps\\_lat,gps\\_long&min=1&max=1000&where=gps\\_fix=3 and tm>'2019-02-14 23:00:19' and tm<'2019-02-15 00:00:00'](http://iot.inteligentny.org/IoT/que.php?func=imeijson&imei=356345080018095&field=ain5,ain6,gps_lat,gps_long&min=1&max=1000&where=gps_fix=3 and tm>'2019-02-14 23:00:19' and tm<'2019-02-15 00:00:00')

#### Query Results:

```
[{ "ain5":"66","ain6":"68","gps_lat":"5202.7326N","gps_long":"02115.8073E","tm":"2019-02-14 23:04:31" },
{ "ain5":"67","ain6":"76","gps_lat":"5202.7328N","gps_long":"02115.8075E","tm":"2019-02-14 23:05:42" },
{ "ain5":"63","ain6":"77","gps_lat":"5202.7328N","gps_long":"02115.8074E","tm":"2019-02-14 23:06:05" },
{ "ain5":"58","ain6":"77","gps_lat":"5202.7328N","gps_long":"02115.8075E","tm":"2019-02-14 23:06:32" },
{ "ain5":"58","ain6":"68","gps_lat":"5202.7328N","gps_long":"02115.8076E","tm":"2019-02-14 23:06:55" } ]
```

### 12.3.3. Odczyt pojedynczego pola z aktualnych statusów z ograniczeniem wyników dla wszystkich urządzeń.

Ta funkcja zwraca ograniczone dane z tabeli „\_ithings”

<http://iot.inteligentny.org/IoT/que.php?func=fieldjson&field=ain5&min=13&max=5000>

#### Parametry:

**func** - fieldjson

**field** - pole do wyświetlenia w wynikach - imei i tm są automatycznie dodawane

**min** - minimalna wartość pola

**max** - maksymalna wartość pola

Dla powyższego ciągu **URL** zwraca wyniki pól **ain5, imei, tm**:

jeśli **ain5** jest w zasięgu (13, 5 000)

#### Query Results:

```
[ { "imei": "353080090069142", "tm": "2019-03-14 11:51:01", "ain5": "14" },  
  { "imei": "356345080018095", "tm": "2019-02-20 09:13:04", "ain5": "115" },  
  { "imei": "karczew", "tm": "2019-03-07 13:08:22", "ain5": "103" } ]
```