

@City Platforme Cloud IoT, CIoT, IIoT



iSys - Intelligent Systems Solutions IoT

@City

@AirQ

@Bin

@Light @Metering @Trace

EN.iSys.PL

Table of Contents

1. Introduction.....	5
1.1 Types d'appareils pris en charge.....	5
1.2. Types de produits pris en charge.....	5
1.3. Protocoles de communication pris en charge.....	5
1.4. Technologie de communication prise en charge des périphériques.....	6
1.5 Serveur Cloud @City	6
1.5.1. Passerelles serveur et de communication.....	7
1.5.2. Interface front-end.....	7
1.5.3. Droits d'accès au serveur.....	8
1.6. Les appareils intelligentes	8
1.6.1. CIoT - appareils GSM.....	8
1.6.2. Systemes de gestion de maison (BMS), IoT - périphériques Ethernet et WiFi.....	8
1.7. Options inter entreprises (B2B).....	9
2. Fonctionnalité de la plateforme @City IoT.....	10
3. Page principale.....	10
4. Formulaire principal.....	10
4.1. Entête.....	11
4.1.1. Lien d'accueil - (ouvre le tableau des résultats réels).....	11
4.1.2. Case à cocher 'X' - ouvre/ferme le formulaire de requête.....	11
4.1.3. Case à cocher 'V' - ouvre/ferme le formulaire Fields.....	11
4.1.4. Icônes graphiques - liens vers les résultats de visualisation (modifiables).....	11
4.2. Forme:.....	12
4.2.1. Case à cocher 'X' - ouvre/ferme tout le formulaire de requête.....	12
4.2.2. CSS - Sélectionnez un thème de visualisation.....	12
4.2.3 Case à cocher Champs visibles – Affiche/masque la liste de filtres de champs.....	12
4.2.4. Onglet: Nom de l'onglet à ajouter ou à supprimer.....	12
4.2.5. Ajouter/Supprimer des boutons - Ajouter ou supprimer des onglets avec le nom dans le champ Onglet.....	12
4.2.6. Bouton de sélection.....	12
4.2.7. Bouton Désélectionner tout.....	12
4.2.7. Bouton Sélectionner tout.....	12
4.2.8. Masquer le filtre - Masquer le formulaire entier.....	12
4.2.9. Bouton Exécuter - Modifier les paramètres.....	12
4.2.10. Case à cocher «V» - affiche/filtre les champs élevés.....	12
4.3. Onglets.....	12
4.4. Contenu de la table.....	13
4.4.1. Exécuter - affiche le type de résultat.....	13
4.4.2. Copier (+/- liens).....	13
4.4.3. Liens de cellules de table.....	13
4.5 Ordre de données.....	13
4.6. Exemple.....	13

5. Cartes.....	14
5.1. Initialisation de la carte.....	14
5.2. Paramètres facultatifs pour la requête.....	15
5.2.1. Modify MAP scale: Modifier l'échelle de la carte (niveau de zoom).....	15
5.2.2. IMEI (champ de sélection du périphérique).....	15
5.2.3. Lon, Lat (champs de coordonnées Longitude, Latitude).....	15
5.2.4. Modify MAP Style: Modifier le style de la carte (thème).....	15
5.2.5. Clause Where.....	15
5.2.6. Exécute (bouton Exécuter la requête).....	16
5.2.7. Deselect All (Supprimer tous les champs de la requête).....	16
Après avoir appuyé sur ce bouton, au moins un champ doit être sélectionné manuellement pour afficher les résultats sur la carte.....	16
5.2.8. Case à cocher 'V' (formulaire Ouvrir/Fermer un champ).....	16
5.2.9. Case à cocher 'X' (Afficher/Masquer le formulaire de requête).....	16
5.3. Exemple.....	16
6. Afficher les résultats dans le tableau.....	18
6.1. Initialisation de la table.....	18
6.2. Options des Paramètres pour la requête.....	19
6.2.1. Sort - trier le champ et ordre croissant/décroissant.....	19
6.2.2. DB/IMEI - Sélection du périphérique.....	19
6.2.3. CSS - style de sélection (thème de visualisation).....	20
6.2.4. Visible Fields: Champs visibles - Formulaire Afficher/Masquer les champs.....	20
6.2.5. Remove Empty: Supprimer le vide - Ne pas afficher les colonnes vides.....	20
6.2.6. Case à cocher 'X' (Afficher/Masquer le formulaire de requête).....	20
6.2.7. Clause Where (pour la limitation des données).....	20
6.2.8. Bouton Select Core (Activer les champs les plus communs).....	20
6.2.9. Bouton Deselect All (Supprimer tous les champs de la requête).....	20
6.2.10. Exécuter (bouton Exécuter la requête).....	20
6.2.11. Case à cocher 'V' (formulaire Ouvrir/Fermer un champ).....	21
7. Graphiques à barres.....	21
8. Graphiques historiques.....	22
8.1. Initialisation des graphiques historiques.....	22
8.2. Paramètres facultatifs des graphiques historiques.....	23
8.2.1. IMEI - (Sélectionnez le périphérique pour afficher les données historiques).....	23
8.2.2. Min - limite la valeur minimale du premier champ.....	23
8.2.3. Max - limite de la valeur maximale du premier champ.....	23
8.2.4. 'V' - Formulaire Afficher/Masquer les champs.....	23
8.2.5. From: définir la date/heure minimale (*).....	23
8.2.6. To: définir la date/heure maximale (*).....	23
8.2.7. Case à cocher 'X' (Afficher/Masquer le formulaire de requête).....	23
8.2.8. Clause «Where».....	23
8.2.9. Bouton Deselect All : Désélectionner tout (Supprimer tous les champs de la requête).....	24
8.2.10. Exécute (bouton Exécuter la requête).....	24
8.2.11. Case à cocher 'V' (formulaire Ouvrir/Fermer un champ).....	24

8.3. Variante de barres: (affiche uniquement les données disponibles).....	24
8.4. Variante continue (avec les mêmes données):.....	25
9. Compatibilité navigateur Web.....	25
10. Personnalisation des thèmes.....	26
11. Mise à jour des algorithmes.....	27
12. Structure de la base de données.....	28
12.1. Structure de tableaux 'ithings_' and '*'	29
12.2. Liste des commandes de périphériques (Events) queue '*_c' table – structure.....	30
12.3. Accès aux résultats des bases de données - Niveau intermédiaire (lecture de données).....	30
12.3.1. Obtenir les statuts actuels de tous les appareils.....	30
12.3.2. Obtenir des données historiques pour le périphérique.....	31
12.3.3. Obtenir la liste des périphériques - un seul champ à partir des états actuels avec limitation.....	32

1. Introduction.

@City IoT Platforme Cloud est un système «micro-cloud» dédié destiné aux clients individuels. La plateforme n'est pas partageable et un seul client a accès à un serveur physique ou virtuel (VPS ou serveurs dédiés). Le client peut sélectionner l'un des dizaines de centres de données en Europe ou dans le monde.

1.1 Types d'appareils pris en charge.

La plateforme @City IoT est dédiée aux produits iSys.PL suivants

- IoT (Internet of things, internet des objets connectées)
- CIoT (Internet des Objets mobiles/cellulaire connecté)
- WiFi
- Ethernet

1.2. Types de produits pris en charge.

La plateforme Cloud IoT @City (eCity) est un système de tailles différentes pour les produits IP IoT (appelés ensemble en tant que périphériques @City ou CioT):

- @City Ville Intelligente
- @Light Eclairage Intelligente
- @Metering Compteurs intelligentes
- @Trace Tracabilité intelligente
- @AirQ System Environmental Intelligente
- @ Bin Poubelle Intelligente

1.3. Protocoles de communication pris en charge

La plateforme @City IoT prend en charge les protocoles suivants pour la communication:

- UDP - suggéré pour les capteurs/dispositifs CIoT (en particulier NBIoT) - faible utilisation des données
- TCP/IP - suggéré pour les appareils avec communication bidirectionnelle - Handshaking/confirmation
- HTTP - suggéré uniquement pour l'accès aux données/visualisation/exportation/"cloud to cloud"

1.4. Technologie de communication prise en charge des périphériques

La plateforme @City IoT prend en charge:

- GSM: périphériques 2G, 3G, 4G (LTE), CATM1 (LTE-M1), NB-IoT (communication UDP/TCP/IP)
- Contrôleurs Ethernet (communication UDP/TCP)
- Contrôleurs WiFi (communication UDP/TCP)
- Futurs produits IP
- Produits non IP via une passerelle matérielle locale supplémentaire @City/eHouse.PRO

@City IoT Platform est dédié à:

- Produits Native iSys:
 - Devices Périphériques CIoT, IoT, IIoT
 - WiFi - produits dédiés pour @City Platform
 - Ethernet - produits dédiés pour @City Platform
- Produits de coproduction (fabriqués par des tiers sous licence iSys et marque externe)
- Produits franchisés (fabriqués par des tiers sous licence iSys avec les logos iSys)

1.5 Serveur Cloud @City

Le logiciel @City fonctionne sur un VPS (serveur privé virtuel) basé sur Linux ou un serveur dédié du côté Internet, en fonction des performances requises du serveur (appelé serveur ultérieur):

- accès privé/public
- nombre total d'appareils
- mise à jour de la fréquence d'état des appareils
- taux de rafraîchissement des données

Plusieurs variantes de VPS existent selon:

- Prix
- géolocalisation de centre de données
- Cœurs de processeur virtuels (1-8)
- RAM virtuelle (1-32 GB)
- Disque SSD (20GB à 1TB)

Des dizaines de serveurs dédiés existent selon:

- Prix
- géolocalisation de centre de données
- Cœurs de processeur (4 .. 32)
- RAM (16 .. 512GB)
- SSD/HDD (256GB .. 8TB)

La plateforme @City IoT est dédiée à un seul client:

- gouvernements et autorités locales (ville, communauté, pays)
- B2B (pour tiers propriétaire)

Comme il ne s'agit pas d'un serveur partageable entre clients, cela simplifie l'accès à la sécurité et les problèmes de performances. Pour cette raison, seul le client est responsable de l'efficacité de la sécurité, de la stabilité, de l'efficacité, du débit des données, etc. En cas de performances insuffisantes, il peut acheter des forfaits de niveau supérieur (VPS ou Dedicated Server), des fonctionnalités et des performances optimales par rapport aux attentes.

Dans des cas particuliers, la communication «Cloud to Cloud» peut être mise en œuvre pour la globalisation et la centralisation des données vers de plus grandes zones plutôt que vers le cloud multi-clients.

1.5.1. Passerelles serveur et de communication

La communication de @City Server est réalisée sur la base d'une application de bas niveau pour la maximisation des performances.

Les principales caractéristiques de l'application @City Server sont les suivantes:

- recevoir des données de manière asynchrone à partir de périphériques (via les protocoles: UDP, TCP IP, HTTP)
- minimiser l'utilisation des données et leurs coûts entre les périphériques et le Cloud (en utilisant des protocoles de communication de bas niveau UDP, TCP IP)
- recevoir le statut chiffré des appareils (via n'importe quel support de communication)
- autorisation et validation des données des appareils en déchiffrant leurs statuts
- décoder le statut des périphériques et le mettre à jour directement dans les tables MariaDB/MySQL (au format de données brutes)
 - table de données actuelle (contient uniquement le statut le plus récent de chaque périphérique)
 - table de données historiques (contient tous les statuts pour un seul périphérique)
- envoyer des commandes en attente (événements) aux périphériques
- assurer la prise de contact, la vérification et les confirmations

Le logiciel @City Server est identique pour chaque utilisateur et ne peut pas être personnalisé pour différents clients.

1.5.2. Interface front-end

L'interface frontale est réalisée avec des scripts PHP permettant d'extraire des données personnalisées de @City Cloud Database. Il utilise un mécanisme de recherche très élastique, basé sur les requêtes SQL originales pour limiter les données souhaitées. L'interface fournit les résultats de la requête au format JSON pour un décodage et un traitement ultérieurs par "l'application" Web frontale JavaScript. L'interface frontale d'origine est la même pour chaque utilisateur et ne peut pas être personnalisée pour différents clients.

L'interface de superposition peut être créée par notre personnel ou en coopération pour assurer la personnalisation pour le client.

1.5.3. Droits d'accès au serveur

Les droits d'accès du client (au serveur physique) sont limités.

Accès aux fichiers uniquement pour le répertoire 'templates' (fichiers texte natifs - .txt, .js, .css, .html):

- ajouter, supprimer, modifier des fichiers HTML purs (interface graphique front-end/développement d'interface)
- ajouter, supprimer, modifier des fichiers JavaScript purs (interface graphique et développement d'algorithmes Front-End)
- ajouter, supprimer, modifier des fichiers CSS purs (Front-End - Vues/thèmes personnalisés)
- ajouter, supprimer, modifier des fichiers texte de modèles (interface graphique front-end)
- ajouter, supprimer, modifier des onglets, des raccourcis, des liens vers des résultats de données

Autres droits d'accès:

- Accès complet à @City Cloud Database MySQL/MariaDB où toutes les données de périphériques sont stockées
- Les accès aux services Web sont définis (informations d'identification) pour le client.
- Le client n'est pas autorisé à transmettre les informations d'identification à des tiers (un accès multiple aux résultats peut affecter les performances, la stabilité et la sécurité du système global).
- Dans des circonstances spéciales et lors de l'utilisation d'un serveur dédié haute performance, un compte public peut être configuré pour observer les données les plus récentes (non historiques).
- L'utilisateur peut dupliquer des données sur son propre serveur MySQL et effectuer ses propres analyses et traitements, afin de ne pas affecter les performances de @City Server.

iSys - le personnel de Intelligent Systems - dispose d'un accès illimité à l'ensemble du serveur, y compris le compte root et à un accès complet à la base de données pour la maintenance.

Dans certaines circonstances, iSys peut octroyer des droits limités supplémentaires au client (scripts PHP, fichiers) après vérification du code source et exécution de tests, si cela n'affecte pas la sécurité, la stabilité et les performances globales du système.

1.6. Les appareils intelligentes

1.6.1. CIoT - appareils GSM

Nos appareils contiennent un microcontrôleur et un module GSM (2G..4G, NBIoT, CATM1)/GPS/GNSS pour la communication. Microcontroller contient un chargeur de démarrage chiffré pour une mise à niveau sécurisée du micrologiciel OTA. Cela permet de créer de nombreuses variantes du système basées sur le même «périphérique intelligent CIoT».

1.6.2. Systemes de gestion de maison (BMS), IoT - périphériques Ethernet et WiFi

Les contrôleurs Ethernet et WiFi permettent une communication IP vers le système (sans facturer le transfert de données à l'opérateur GSM). Ces périphériques ont également un chargeur de démarrage chiffré et les périphériques peuvent être mis à jour via son interface native. Pour le WiFi, mise à niveau du micrologiciel OTA à partir du serveur principal.

1.7. Options inter entreprises (B2B)

Il existe plusieurs options pour les affaires et la coopération:

- produits finis pour imports/exports (standards, appareils) - produits finis
- importation/exportation de produits OEM (PCB, contrôleurs, etc.) - produits intermédiaires, pièces de rechange
- franchise - production basée sur notre licence pour les marchés locaux (nous fournissons uniquement des microcontrôleurs programmés) sous iSys - Marques de systèmes intelligents
- Coproduction - (voir ci-dessus) mais sous marques externes

2. Fonctionnalité de la plateforme @City IoT

La plate-forme @City prend en charge un modèle Front-End personnalisable pour la visualisation, l'interrogation, la limitation et le traitement de données (données actuelles/historiques):

- visualiser les données sélectionnées sur la carte (géolocalisation)
- montre les données sélectionnées et les résultats dans les tableaux
- affiche les données sélectionnées dans des graphiques à barres
- montre les données sélectionnées dans les graphiques historiques
- exporter des données vers des applications tierces et les traiter ultérieurement
- d'autres fonctionnalités seront mises à jour sur les demandes individuelles.

Le serveur frontal de l'utilisateur est accessible via un domaine/sous-domaine/fichier/adresse IP statique ou redirection DNS, le cas échéant.

Exemplary & Demo installation (Il est activé uniquement pour les clients potentiels).

Veuillez nous informer lorsque vous souhaitez le tester - pour permettre un accès public à la plate-forme.

Cela pourrait nécessiter une adresse IP statique de l'ordinateur distant pour permettre la communication avec la plate-forme @City.

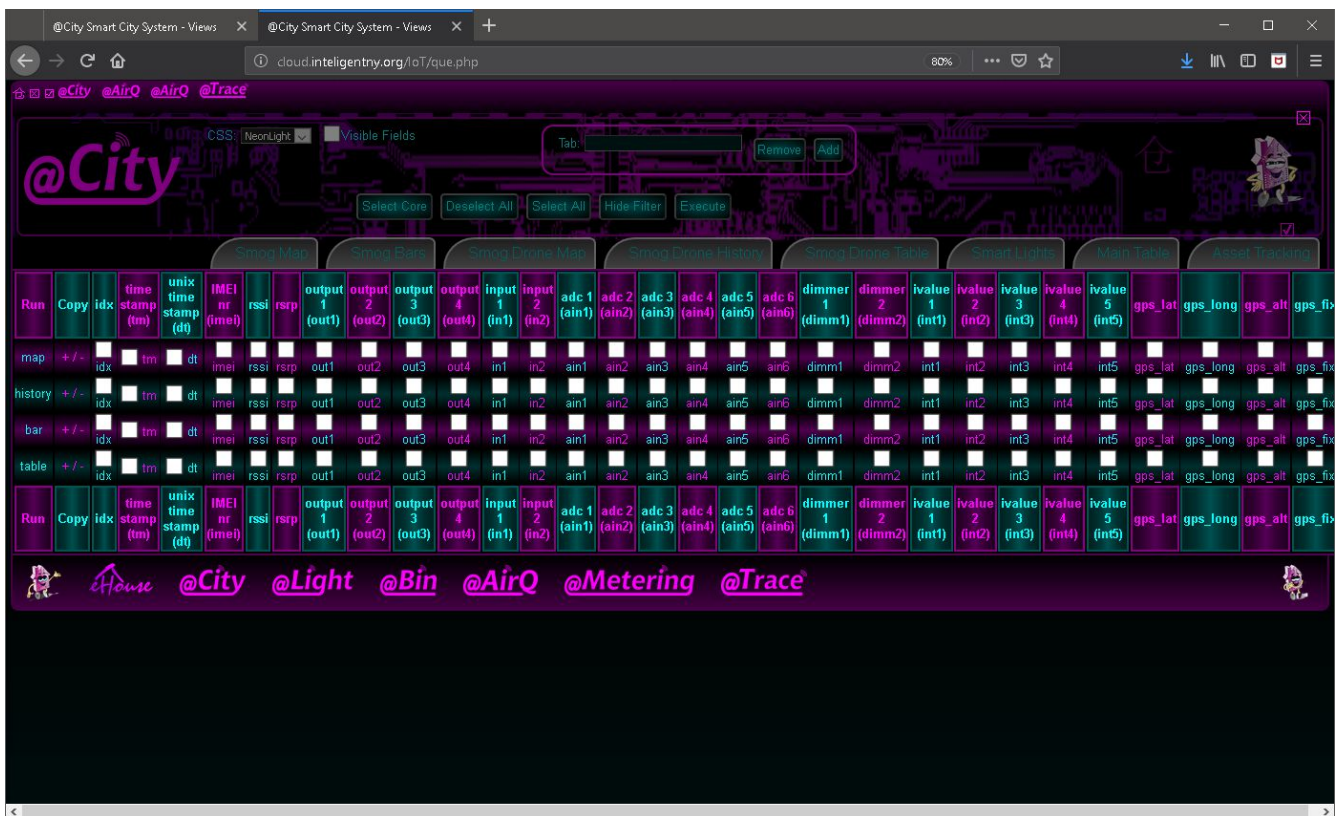
3. Page principale

La page principale est laissée volontairement vide pour des raisons de sécurité: <http://%YourIP%/IoT/>

Il peut être activé et modifié individuellement et contenir des liens vers tous les services disponibles de @City IoT Platform si nécessaire.

4. Formulaire principal

Main Form est destiné à créer de nouveaux presets et onglets: <http://iot.intelligentny.org/IoT/que.php>



C'est la forme initiale pour créer des résultats, des vues et des onglets pour chaque configuration

Descriptions (de haut en bas et de droite à gauche)

4.1. Entête

4.1.1. Lien d'accueil - (ouvre le tableau des résultats réels)

4.1.2. Case à cocher 'X' - ouvre/ferme le formulaire de requête

4.1.3. Case à cocher 'V' - ouvre/ferme le formulaire Fields

4.1.4. Icônes graphiques - liens vers les résultats de visualisation (modifiables)

4.2. Forme:

4.2.1. Case à cocher 'X' - ouvre/ferme tout le formulaire de requête

4.2.2. CSS - Sélectionnez un thème de visualisation

Modifier le thème de visualisation Le fichier CSS doit exister dans le répertoire «templates/css/» - répertorié automatiquement.

4.2.3 Case à cocher Champs visibles – Affiche/masque la liste de filtres de champs

4.2.4. Onglet: Nom de l'onglet à ajouter ou à supprimer

4.2.5. Ajouter/Supprimer des boutons - Ajouter ou supprimer des onglets avec le nom dans le champ Onglet

4.2.6. Bouton de sélection

Sélectionnez les principaux champs visibles sur la table. Il est mis à jour automatiquement.

4.2.7. Bouton Désélectionner tout

Désélectionner tous les champs (doit être suivi d'une sélection manuelle de certains d'entre eux)

4.2.7. Bouton Sélectionner tout

Sélectionner tous les champs (il faut ensuite désélectionner certains d'entre eux manuellement)

4.2.8. Masquer le filtre - Masquer le formulaire entier

Ceci est équivalent à toutes les cases (X).

4.2.9. Bouton Exécuter - Modifier les paramètres

4.2.10. Case à cocher «V» - affiche/filtre les champs élevés.

4.3. Onglets

Onglets créés individuellement avec des noms et des préreglages (stockés dans le fichier cfg/tabs.cfg). Le fichier contient en fait un nom et une URL (séparés par un caractère de tabulation).

4.4. Contenu de la table

Affiche tous les champs limités par le filtre de champ.

Champs de la table:

4.4.1. Exécuter - affiche le type de résultat

map - résultats cartographiques sur la carte (un ou plusieurs champs peuvent être sélectionnés)

history - cartes historiques (un ou plusieurs champs peuvent être sélectionnés)

onglet - affiche le tableau (toute combinaison de champs peut être sélectionnée)

barre - un seul champ est affiché sur l'histogramme

En appuyant sur l'une de ses valeurs, de nouveaux résultats s'affichent avec les champs sélectionnés (pour la ligne actuelle).

4.4.2. Copier (+/- liens)

Ajouter/supprimer un onglet avec le nom défini dans le champ Onglet. Il utilise uniquement les champs sélectionnés dans la même ligne de la table.

4.4.3. Liens de cellules de table

Appuyez sur un autre nom de champ pour lancer la visualisation des données du champ sélectionné pour la ligne sélectionnée.

4.5 Ordre de données

L'ordre des champs affichés est le même que son ordre dans le champ des champs (toutefois, le champ tm est toujours envoyé à la fin du texte). Cet ordre ne peut être modifié qu'avec l'édition directe des paramètres d'URL (partie d'ordre des champs).

4.6. Exemple

Par exemple: onglet Paramétrage avec le nom de suivi des actifs et contenant la carte avec le temps et la vitesse sur la carte

Toute description faisant référence à la ligne où le texte «Map» est dans la colonne «Exécuter».

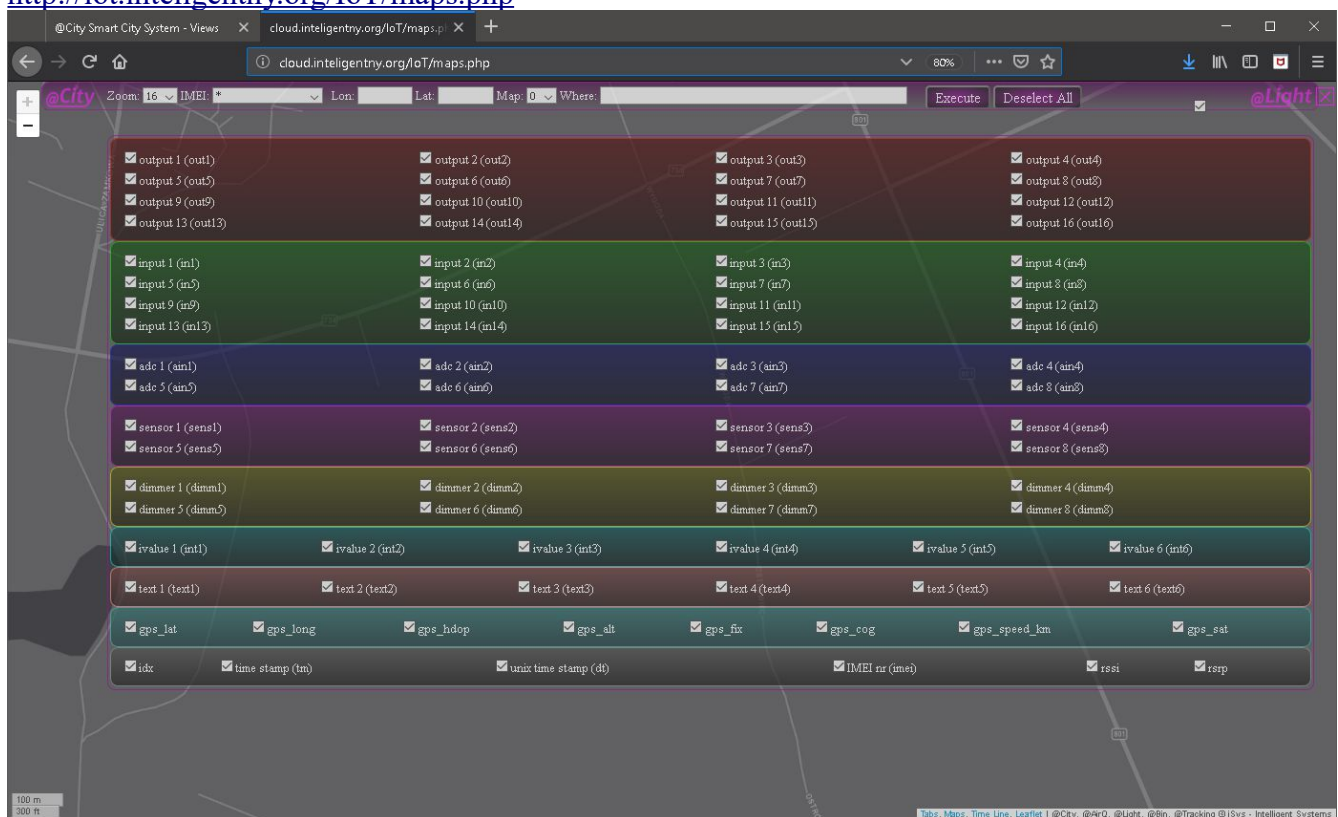
- 1) Entrez le nom «Asset Tracking» dans le champ Tap (sans les guillemets)
- 2) Assurez-vous que toutes les colonnes ne sont pas sélectionnées dans la ligne.
- 3) sélectionnez tm, gps_speed_km uniquement dans la ligne
- 4) appuyez sur le bouton + où dans la ligne

5. Cartes

Les cartes peuvent être lancées à partir de MainForm avec une préconfiguration

5.1. Initialisation de la carte

L'initialisation de la carte est effectuée manuellement lorsqu'elle est exécutée directement avec le lien:
<http://iot.intelligentny.org/IoT/maps.php>



1. L'utilisateur doit désélectionner tous les champs (appuyez sur le bouton Désélectionner).
2. Cochez les cases des champs affichés (par exemple, Ain5 (niveau de smog) et tm (date/heure de la mesure)).
3. Cochez la case "V" pour masquer les champs
4. Appuyez sur le bouton Execute pour exécuter la requête de base de données et afficher les informations actuelles de tous les capteurs/périphériques.
5. La carte avec les données est mise à jour après 30 secondes ou plus.

5.2. Paramètres facultatifs pour la requête

Paramètres décrits de gauche à droite (sur la capture d'écran ci-dessus).

5.2.1. Modify MAP scale: Modifier l'échelle de la carte (niveau de zoom)

1) Le niveau de zoom peut être modifié à l'aide des boutons (+/-) d'échelle ($\text{current_scale} \times 2$ ou $\text{current_scale} / 2$ respectivement). Appuyez sur l'un de ces boutons pour modifier automatiquement l'échelle.

2) Vous pouvez également sélectionner Niveau de zoom dans le champ Zone de liste déroulante Zoom et appuyer sur le bouton Exécuter. Dans ce cas, l'ensemble de la vue/carte est rechargé et actualisé (prend un certain temps lors de l'initialisation).

5.2.2. IMEI (champ de sélection du périphérique)

Le champ IMEI contient l'ID unique du périphérique ou l'alias unique d'un périphérique. Le paramètre par défaut est * (astérisque), qui indique les valeurs les plus récentes et la géolocalisation de chaque périphérique.

Si vous définissez une autre valeur pour IMEI, les données d'historique du périphérique sélectionné seront affichées. Cela n'a de sens que pour les capteurs mobiles et mobiles, sinon les résultats se chevaucheront sur la carte au même endroit.

5.2.3. Lon, Lat (champs de coordonnées Longitude, Latitude)

Définir la position centrale de la carte. Ce champ est défini sur la position du curseur lorsque vous appuyez sur le bouton de la souris sur la carte.

5.2.4. Modify MAP Style: Modifier le style de la carte (thème)

Le style/thème de la carte peut être sélectionné dans le champ ComboBox de la carte (par exemple, foncé, gris, topographique).

Divers thèmes de carte peuvent avoir différents niveaux de zoom maximal, ce qui permet d'appliquer le thème approprié pour augmenter l'échelle de la carte.

5.2.5. Clause Where

La clause Where est utilisée pour la chaîne de requête supplémentaire {partie WHERE} pour MySQL/MariaDB.

Cette clause est prise en compte pour la chaîne complète QUERY de construction complète pour le

résultat de la base de données. Il peut limiter les données, le temps et toute autre valeur en limitant le nombre de résultats. Les noms de champs de table originaux (pas d'alias) doivent être utilisés dans ce champ. Par exemple.

1. `gps_speed_km > 10` // la vitesse est supérieure à 10km/h
2. `ain5 > 3` // ain5 est supérieur à 3 (avec un nombre de particules de 2,5µm - niveau de smog)
3. `gps_speed_km > 10` et `ain6 > 5` // la vitesse est supérieure à 10km/h et ain6 est supérieure à 5 (avec un nombre de particules de 10µm - niveau de smog)

5.2.6. Exécute (bouton Exécuter la requête)

Il est nécessaire d'appuyer sur ce bouton pour modifier les paramètres et les paramètres (à l'exception des boutons +/-).

La carte est chargée depuis le début avec de nouveaux préréglages.

La carte n'est pas chargée du tout, lorsqu'aucune donnée n'est disponible pour la requête en cours.

5.2.7. Deselect All (Supprimer tous les champs de la requête)

Après avoir appuyé sur ce bouton, au moins un champ doit être sélectionné manuellement pour afficher les résultats sur la carte.

5.2.8. Case à cocher 'V' (formulaire Ouvrir/Fermer un champ)

Cette case à cocher est utilisée pour afficher/masquer le sélecteur de champs à afficher.

5.2.9. Case à cocher 'X' (Afficher/Masquer le formulaire de requête)

Cette case à cocher permet de masquer tout le formulaire sauf (boutons +/-)

Les résultats sur la carte sont continuellement actualisés et mis à jour avec de nouvelles valeurs

5.3. Exemple

Par exemple, résultats de smog (capteur installé sur la voiture): Particules de niveau de smog 2,5 µm (Ain5), vitesse (`gps_speed_km`), date/heure (tm), carte (2 - topographique), niveau de zoom 16,

Clause Where:

"Gps_fix = 3 et tm> '2019-02-18 00:00:00' et tm <'2019-02-19 00:00:00' et gps_speed_km> 0".

// GPS = résultats 3D valides & date = 2019-02-18 & vitesse> 0km/h

6. Afficher les résultats dans le tableau

Afficher les résultats dans le tableau.

Sur «Main Form», appuyez sur «Table», après avoir sélectionné certains champs pour afficher le tableau préconfiguré.

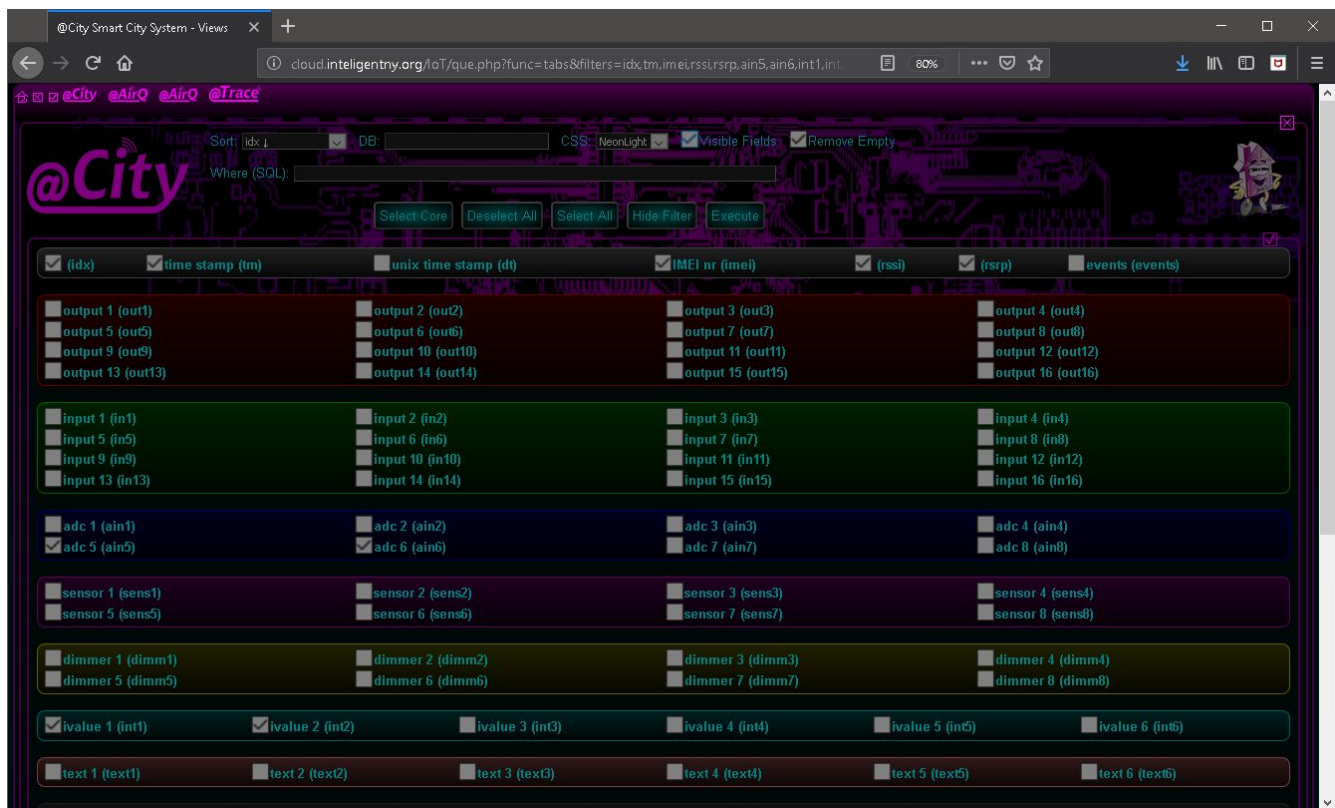
The screenshot shows the @City IoT Cloud Platform interface. At the top, there's a navigation bar with icons for @City, @AirQ, and @Trace. Below this is a search bar and a table of data. The table has columns for idx, time stamp (tm), IMEI nr (imei), rssi, rsrp, adc 5 (ain5), adc 6 (ain6), ivalue 1 (int1), ivalue 2 (int2), gps_lat, gps_long, gps_alt, gps_fix, gps_speed_km, and gps_sat. The data rows show various sensor readings over time.

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	2019-03-12 21:10:17	353080090069142	72	1	45	93	19	115	5202.7354N	02115.8172E	86.6	3	0.0	07
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 13:08:22	karczew	76	76	103	62	19	77	5203.9312N	02115.4106E	88.2	3	0.0	09

6.1. Initialisation de la table

Lorsque la table est ouverte à partir du lien <http://iot.inteligeny.org/IoT/que.php?func=tabs>, elle nécessite une initialisation préalable des paramètres.

Vous pouvez sélectionner les champs visibles (en appuyant sur la case «Champs visibles»).



1. Cochez toutes les cases requises pour les champs affichés
2. Cochez la case «Champs visibles» pour masquer le formulaire.
3. Appuyez sur le bouton Execute pour exécuter la requête de la base de données et afficher la table.

6.2. Options des Paramètres pour la requête

Les paramètres sont décrits de gauche à droite (sur la capture d'écran).

6.2.1. Sort - trier le champ et ordre croissant/décroissant

Le champ de tri équivaut à appuyer sur l'en-tête de colonne.

6.2.2. DB/IMEI - Sélection du périphérique

Le champ IMEI contient l'ID unique du périphérique ou l'alias unique d'un périphérique. Avec une valeur vide, il affiche le tableau des valeurs les plus récentes.

Si vous définissez une autre valeur pour IMEI, les données d'historique du périphérique sélectionné

seront affichées.

6.2.3. CSS - style de sélection (thème de visualisation)

6.2.4. Visible Fields: Champs visibles - Formulaire Afficher/Masquer les champs

6.2.5. Remove Empty: Supprimer le vide - Ne pas afficher les colonnes vides

6.2.6. Case à cocher 'X' (Afficher/Masquer le formulaire de requête)

6.2.7. Clause Where (pour la limitation des données)

Ceci est un suffixe pour la chaîne de requête supplémentaire MySQL/MariaDB {WHERE part}

Cette clause est prise en compte pour construire une chaîne QUERY complète pour le résultat de la base de données. Il peut limiter les données, le temps et toute autre valeur en limitant le nombre de résultats. Les noms de champs de table originaux (pas d'alias) doivent être utilisés dans ce champ. Par exemple.

1. gps_speed_kmh > 10 // la vitesse est supérieure à 10km/h
2. ain5 > 3 // ain5 est supérieur à 3 (avec un nombre de particules de 2,5µm - niveau de smog)
3. gps_speed_kmh > 10 et ain6 > 5 // la vitesse est supérieure à 10km/h et ain6 est supérieure à 5 (avec un nombre de particules de 10µm - niveau de smog)

6.2.8. Bouton Select Core (Activer les champs les plus communs)

6.2.9. Bouton Deselect All (Supprimer tous les champs de la requête)

Après avoir appuyé sur ce bouton, au moins un champ doit être sélectionné manuellement pour afficher les résultats sur la carte.

6.2.10. Exécuter (bouton Exécuter la requête)

Il est nécessaire d'appuyer sur ce bouton pour modifier les paramètres et les paramètres (à l'exception

des boutons +/-).

La table est rechargée depuis le début avec de nouveaux préréglages.

6.2.11. Case à cocher 'V' (formulaire Ouvrir/Fermer un champ)

Cette case à cocher est utilisée pour afficher/masquer le sélecteur de champs à afficher.

Les résultats de la table sont triés en fonction du paramétrage du champ Trier. L'ordre de tri peut être modifié en appuyant sur l'en-tête de la ligne (une fois pour une direction, deux fois pour une autre direction).

Certains résultats dans les colonnes sont liés à d'autres écrans de visualisation (codés en dur).

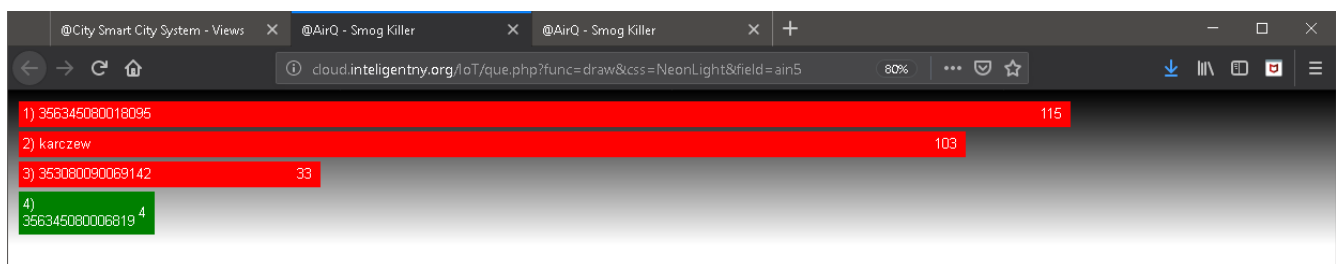
Lorsque vous affichez des données historiques pour un périphérique, vous devez les limiter afin de ne pas afficher des informations historiques complètes, car cela pourrait entraîner des problèmes de performances ou de mémoire insuffisante.

7. Graphiques à barres.

Les graphiques à barres doivent être exécutés à partir du formulaire principal en appuyant sur un seul champ dans la ligne "Bar".

Il affiche les barres triées normalisées à la valeur maximale, en commençant par l'ordre le plus élevé.

C'est utile pour vérifier rapidement les résultats extrêmes et prendre des mesures.



Un survol de la souris affichera des informations supplémentaires sur le périphérique.

8. Graphiques historiques.

Les graphiques historiques peuvent être lancés à partir de MainForm en appuyant sur la colonne sélectionnée dans la ligne 'History' (pour un seul champ).

Pour plusieurs champs dans la ligne "History", les champs souhaités doivent être cochés et le lien "History" doit être activé dans la colonne "Exécute".

Les résultats historiques sont limités aux dernières 24 heures et aux 24 prochaines heures (pour les graphiques éventuellement rafraîchissants), lorsqu'aucune limite n'a été définie.

8.1. Initialisation des graphiques historiques

Les graphiques historiques, lorsqu'ils sont ouverts à partir d'un lien principal, nécessitent une initialisation sous la forme d'autres résultats, lorsqu'ils sont ouverts à partir de paramètres de lien sans préférences.

Plusieurs champs peuvent être sélectionnés pour afficher divers éléments. Il peut également être défini dans le formulaire de filtre de champ.

The screenshot shows the '@City' web interface. At the top, there's a navigation bar with '@City', '@AirQ', and '@Trace'. Below it, a search bar and a 'Where (SQL)' input field are visible. A grid of checkboxes allows selecting various data fields. The fields are organized into rows and columns, with some already selected. The interface also features buttons for 'Select Core', 'Deselect All', 'Select All', 'Hide Filter', and 'Execute'.

Field Type	Field Name	Selected
Output	output 1 (out1)	<input type="checkbox"/>
Output	output 2 (out2)	<input type="checkbox"/>
Output	output 3 (out3)	<input type="checkbox"/>
Output	output 4 (out4)	<input type="checkbox"/>
Output	output 5 (out5)	<input type="checkbox"/>
Output	output 6 (out6)	<input type="checkbox"/>
Output	output 7 (out7)	<input type="checkbox"/>
Output	output 8 (out8)	<input type="checkbox"/>
Output	output 9 (out9)	<input type="checkbox"/>
Output	output 10 (out10)	<input type="checkbox"/>
Output	output 11 (out11)	<input type="checkbox"/>
Output	output 12 (out12)	<input type="checkbox"/>
Output	output 13 (out13)	<input type="checkbox"/>
Output	output 14 (out14)	<input type="checkbox"/>
Output	output 15 (out15)	<input type="checkbox"/>
Output	output 16 (out16)	<input type="checkbox"/>
Input	input 1 (in1)	<input type="checkbox"/>
Input	input 2 (in2)	<input type="checkbox"/>
Input	input 3 (in3)	<input type="checkbox"/>
Input	input 4 (in4)	<input type="checkbox"/>
Input	input 5 (in5)	<input type="checkbox"/>
Input	input 6 (in6)	<input type="checkbox"/>
Input	input 7 (in7)	<input type="checkbox"/>
Input	input 8 (in8)	<input type="checkbox"/>
Input	input 9 (in9)	<input type="checkbox"/>
Input	input 10 (in10)	<input type="checkbox"/>
Input	input 11 (in11)	<input type="checkbox"/>
Input	input 12 (in12)	<input type="checkbox"/>
Input	input 13 (in13)	<input type="checkbox"/>
Input	input 14 (in14)	<input type="checkbox"/>
Input	input 15 (in15)	<input type="checkbox"/>
Input	input 16 (in16)	<input type="checkbox"/>
ADC	adc 1 (ain1)	<input type="checkbox"/>
ADC	adc 2 (ain2)	<input type="checkbox"/>
ADC	adc 3 (ain3)	<input type="checkbox"/>
ADC	adc 4 (ain4)	<input type="checkbox"/>
ADC	adc 5 (ain5)	<input checked="" type="checkbox"/>
ADC	adc 6 (ain6)	<input checked="" type="checkbox"/>
ADC	adc 7 (ain7)	<input type="checkbox"/>
ADC	adc 8 (ain8)	<input type="checkbox"/>
Sensor	sensor 1 (sens1)	<input type="checkbox"/>
Sensor	sensor 2 (sens2)	<input type="checkbox"/>
Sensor	sensor 3 (sens3)	<input type="checkbox"/>
Sensor	sensor 4 (sens4)	<input type="checkbox"/>
Sensor	sensor 5 (sens5)	<input type="checkbox"/>
Sensor	sensor 6 (sens6)	<input type="checkbox"/>
Sensor	sensor 7 (sens7)	<input type="checkbox"/>
Sensor	sensor 8 (sens8)	<input type="checkbox"/>
Dimmer	dimmer 1 (dim1)	<input type="checkbox"/>
Dimmer	dimmer 2 (dim2)	<input type="checkbox"/>
Dimmer	dimmer 3 (dim3)	<input type="checkbox"/>
Dimmer	dimmer 4 (dim4)	<input type="checkbox"/>
Dimmer	dimmer 5 (dim5)	<input type="checkbox"/>
Dimmer	dimmer 6 (dim6)	<input type="checkbox"/>
Dimmer	dimmer 7 (dim7)	<input type="checkbox"/>
Dimmer	dimmer 8 (dim8)	<input type="checkbox"/>
Value	ivalue 1 (int1)	<input checked="" type="checkbox"/>
Value	ivalue 2 (int2)	<input checked="" type="checkbox"/>
Value	ivalue 3 (int3)	<input type="checkbox"/>
Value	ivalue 4 (int4)	<input type="checkbox"/>
Value	ivalue 5 (int5)	<input type="checkbox"/>
Value	ivalue 6 (int6)	<input type="checkbox"/>
Text	text 1 (text1)	<input type="checkbox"/>
Text	text 2 (text2)	<input type="checkbox"/>
Text	text 3 (text3)	<input type="checkbox"/>
Text	text 4 (text4)	<input type="checkbox"/>
Text	text 5 (text5)	<input type="checkbox"/>
Text	text 6 (text6)	<input type="checkbox"/>

1. Cochez toutes les cases requises pour les champs affichés
2. Cochez la case «Champs visibles» pour masquer le formulaire.
3. Appuyez sur le bouton Execute pour exécuter la requête de base de données et afficher le tableau.

8.2. Paramètres facultatifs des graphiques historiques

Articles décrits en haut et de gauche à droite (sur la capture d'écran).

8.2.1. IMEI - (Sélectionnez le périphérique pour afficher les données historiques)

Le champ IMEI contient l'ID unique du périphérique ou l'alias unique d'un périphérique. Avec * (astérisque), il affiche un tableau des valeurs les plus récentes qui n'a aucun sens.

Si vous définissez une autre valeur pour IMEI, les données d'historique du périphérique sélectionné seront affichées.

8.2.2. Min - limite la valeur minimale du premier champ

8.2.3. Max - limite de la valeur maximale du premier champ

8.2.4. 'V' - Formulaire Afficher/Masquer les champs

8.2.5. From: définir la date/heure minimale (*)

8.2.6. To: définir la date/heure maximale (*)

8.2.7. Case à cocher 'X' (Afficher/Masquer le formulaire de requête)

8.2.8. Clause «Where»

Clause de limitation des résultats de données Chaîne de requête supplémentaire MySQL/MariaDB {partie WHERE}.

Cette clause est prise en compte pour la chaîne complète QUERY de construction complète pour le résultat de la base de données. Il peut limiter les données, le temps et toute autre valeur en limitant le nombre de résultats. Les noms de champs de table d'origine (pas d'alias) doivent être utilisés dans ce champ et une syntaxe SQL valide. Par exemple.

1. `gps_speed_km > 10` // la vitesse est supérieure à 10km/h
2. `ain5 > 3` // ain5 est supérieur à 3 (avec un nombre de particules de 2,5µm - niveau de smog)
3. `gps_speed_km > 10` et `ain6 > 5` // la vitesse est supérieure à 10km/h et ain6 est supérieure à 5 (avec un nombre de particules de 10µm - niveau de smog)

8.2.9. Bouton Deselect All : Désélectionner tout (Supprimer tous les champs de la requête)

Après avoir appuyé sur ce bouton, au moins un champ doit être sélectionné manuellement pour afficher les résultats historiques.

8.2.10. Exécute (bouton Exécuter la requête)

Il est nécessaire d'appuyer sur ce bouton pour modifier les paramètres, les paramètres (sauf l'affichage des champs ou le panneau de requête). La table est rechargée depuis le début avec de nouveaux préreglages.

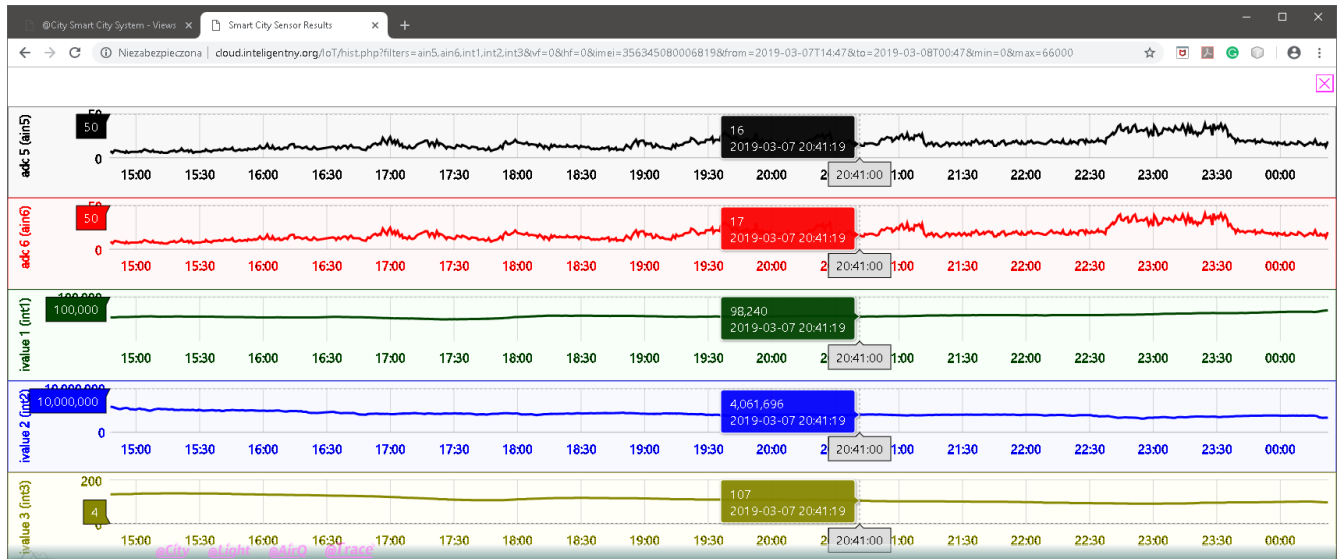
8.2.11. Case à cocher 'V' (formulaire Ouvrir/Fermer un champ)

Cette case à cocher est utilisée pour afficher/masquer le sélecteur de champs à afficher.

8.3. Variante de barres: (affiche uniquement les données disponibles)



8.4. Variante continue (avec les mêmes données):



Le pointeur de la souris affiche les valeurs des mesures et la date/heure.

9. Compatibilité navigateur Web

Function/WWW Browser	Chrome 72	Fire Fox 65	Edge	O
Cartes	+	+	+	+
Historique	+	+	+	+
Bars	+	+	+	+
Onglets	+	+	+	+

* - Firefox ne prend pas en charge le sélecteur de date/heure (le champ de texte doit être modifié manuellement en utilisant le format de date/heure approprié).

Internet Explorer n'est pas pris en charge (utilisez plutôt Edge)

Les autres navigateurs Web n'ont pas été testés. Le pointeur de souris affiche les valeurs de mesure et la date/heure.

10. Personnalisation des thèmes

Les pages Web sont basées sur un fichier de modèle général situé dans le répertoire "modèles" "*.template".

De plus, chaque type de page contient:

- 1) Fichier '*.head' qui stocke l'en-tête de la page (liens, CSS importés, fichiers JavaScript, etc.)
- 2) les fichiers '*.foot' qui stockent le pied de page de la page (liens, etc.)

Le thème de visualisation peut être modifié en fonction des préférences de l'utilisateur en copiant et en modifiant des fichiers CSS. Les fichiers CSS sont situés dans le répertoire 'templates/css'. Différents thèmes de page Web peuvent être utilisés pour créer optimisé pour, par exemple. impression, Smartphones, modèles de PAD.

Vues sous forme de tableau - vous pouvez sélectionner un fichier CSS pour la modification complète du thème dans un champ (stocké dans le répertoire 'templates/css/tabs').

The screenshot shows a web application interface for '@City Smart City System - Views'. It features a navigation bar with icons for home, back, forward, and search. Below the navigation bar, there are tabs for '@City', '@AirQ', '@Light', '@Bin', '@AirQ', '@Metering', and '@Trace'. The main content area displays a data table with columns for 'idx', 'time stamp (tm)', 'IMEI nr (imei)', '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 sensor readings. At the bottom of the interface, there are icons for '@House', '@City', '@Light', '@Bin', '@AirQ', '@Metering', and '@Trace'.

idx	time stamp (tm)	IMEI nr (imei)	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	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
11	2019-03-07 13:08:22	karczew	76	76	0	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	-16777186	60	5202.9500N	02115.8000E	86.5	3	0.0	10
1	2019-02-10 12:56:23	351580051067110	91	99	255	255	-16776767	450	0000.0000N	00000.0000E		4		

Vues de la carte - le thème général est sélectionné par la liste déroulante de type "carte". En outre, le fichier CSS par défaut 'templates/css/map.css' contient des fonctionnalités supplémentaires telles que le masquage/la coloration des résultats en fonction de ses valeurs. Le reste de ce fichier CSS est pratiquement limité aux formulaires de requête et de champ.

La plupart des fichiers PHP @City Platform pour la visualisation acceptent le paramètre css avec la valeur du nom de fichier du thème (sans extension). Le fichier doit être situé dans le répertoire 'templates/css' et son nom est sensible à la casse.

Certains éléments de l'affichage du thème se trouvent directement dans le fichier JavaScript situé dans le répertoire 'template/js'.

Le script principal @City '@City.js' se trouve dans le répertoire supérieur. Il n'y a pas de possibilité de modification à cet emplacement, mais le script peut être copié dans le répertoire 'templates/js' et y être modifié. L'utilisation d'un script individuel nécessite la mise à jour de tous les fichiers d'en-tête.

11. Mise à jour des algorithmes

Certains capteurs uniques peuvent nécessiter des fonctions de calcul dédiées.

Il n'est pas possible de mettre à jour et de gérer plusieurs variantes de @City Server Software, interface frontale PHP, ce qui causerait de nombreux problèmes, versions, erreurs.

Le moyen le plus simple et le plus simple d'y parvenir consiste à mettre à jour les fichiers de superposition en JavaScript pour un affichage correct de la valeur/description.

Les scripts JS d'origine sont des fichiers texte ouverts et peuvent être adaptés aux besoins des clients.

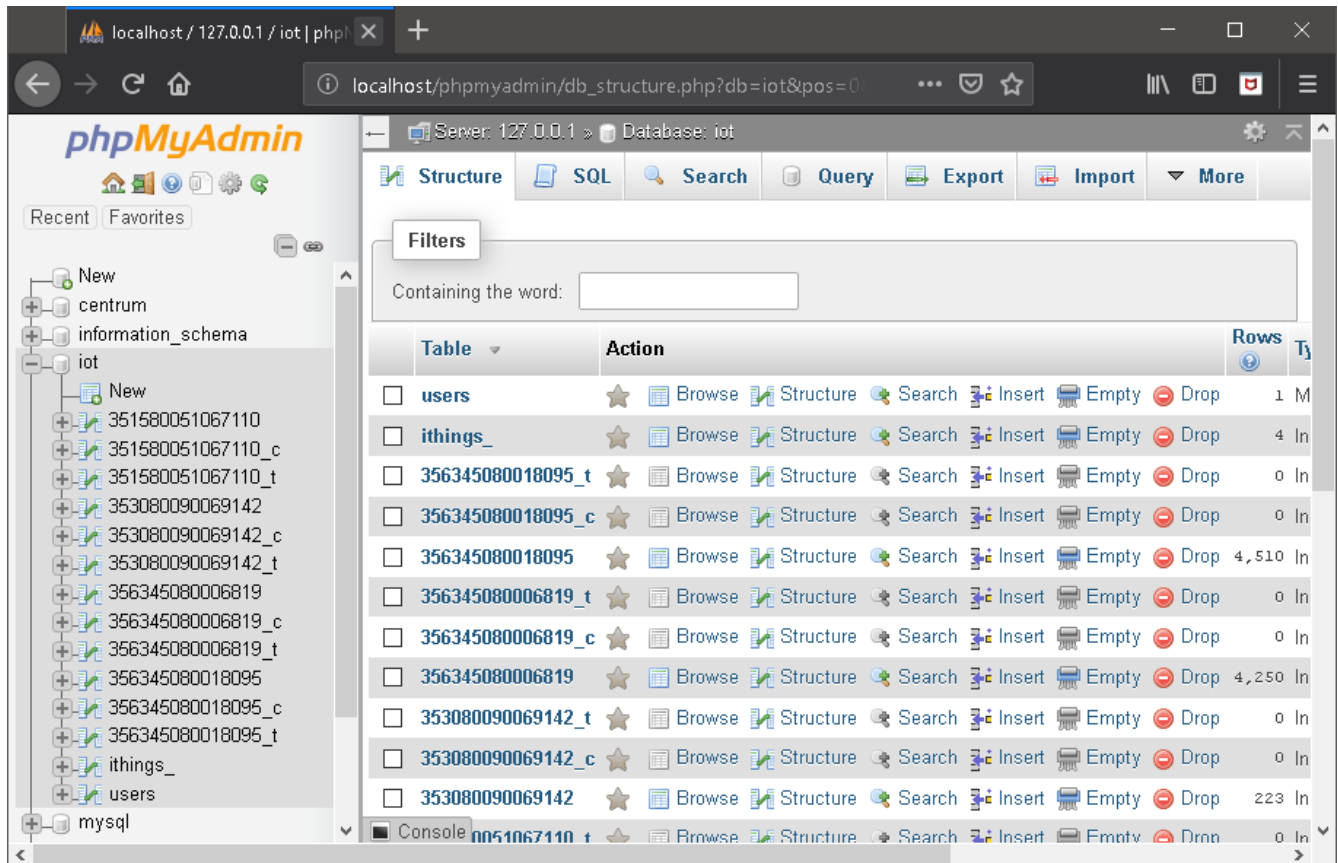
Comme indiqué dans le chapitre précédent, ils doivent être copiés dans le répertoire 'templates/js' où le client dispose des droits d'accès pour la modification.

Les aspects techniques de la programmation du système @City ne sont pas traités dans ce document.

Toutefois, les développeurs Web possédant des connaissances de base en HTML et JS peuvent personnaliser l'application Web frontale en fonction des besoins de chaque client.

12. Structure de la base de données

La base de données @City avec le nom 'IoT' ou '*IoT' est divisée en tables (où astérix est le préfixe dépendant du serveur d'hébergement - si nécessaire). La base de données peut être observée dans PHPAdmin (application Web) à l'adresse lien <http://iot.intelligentny.org/phpmyadmin>



Tables définies pour chaque appareil (where * {asterix} is IMEI address – unique ID):

- '*_t' – pièce pour l'autorisation des utilisateurs actuels (non utilisés pour les clouds mono-client)
- '*_c' – commandes des file d'attente (événements) - à envoyer au périphérique
- '*' – tous les résultats d'état décodés.

Autre tableaux:

- tableau **'ithings_'** - contient le statut actuel décodé de tous les périphériques (il est copié lors de la mise à jour des tables '*'). La structure de la table 'ithings_' est presque identique à celle des tables '*'. 'ithings_' comporte des champs supplémentaires pour l'adresse postale et les descriptions.
- le tableau 'users' est restreint et doit être modifié.

12.1. Structure de tableaux 'ithings_' and '**'

- ``idx`` - indexe
- ``tm`` - horodatage automatique
- ``dt`` - horodatage unix
- ``imei`` - adresse unique de l'appareil
- ``rssi`` - Niveau de signal RSSI
- ``rsrp`` - signal RSRP
- ``events`` - événements en attente à envoyer au contrôleur
- ``out1` .. `out16`` - état des sorties décodées
- ``in1` .. `in16`` - état des entrees décodées
- ``ain1` .. `ain8`` - valeurs ADC décodées (RAW)
- ``sens1` .. `sens8`` - valeurs de capteur converties (dépend du type d'appareil)
- ``dimm1` .. `dimm8`` - valeurs de gradateurs décodés (dépend du type d'appareil)
- ``int1` .. `int6`` - valeurs des compteurs décodés (dépend du type d'appareil)
- ``text1` .. `text6`` - valeurs de texte décodées (dépend du type d'appareil)
- ``creation`` - date/heure de création de l'appareil
- ``last`` - dernier date/heure
- ``user`` - usage futur
- ``pass`` - usage futur
- ``sn`` - numéro de série gsm
- ``status`` - état actuel du contrôleur non décodé au format hexadécimal
- ``hash_code`` - usage futur
- ``addr`` - adresse de périphérique courte
- ``fwnr`` - numéro de firmware
- ``disabled`` - appareil désactivé (non exécuté)
- ``gsm_nr`` - Numéro gsm de la CIoT
- ``vendor`` - code du vendeur (pour la franchise/coproduction)
- ``timezone`` - Décalage horaire
- ``dst`` - Utiliser les paramètres d'économies quotidiennes
- ``gps_lat`` - Latitude GPS
- ``gps_long`` - Longitude GPS
- ``gps_hdop`` - HDOP GPS
- ``gps_alt`` - Altitude GPS
- ``gps_fix`` - Fixation GPS (doit être 3 - pour des résultats valides)
- ``gps_cog`` - GPS Cog
- ``gps_speed_km`` - Vitesse GPS en [km/h]
- ``gps_sat`` - Satellites GPS trouvés
- ``continent``, ``country``, ``region``, ``subregion``, ``subsubregion``, ``city``, ``district``, ``street``, ``street_nr``, ``item_nr`` - **Adresse du client et champs de description (!!!! Non disponible pour les tables historiques '**')**

- `log` - enregistrer les données

Les noms de champs sont importants pour la création de requêtes SQL afin d'accroître la fonctionnalité de recherche, car le nom d'origine doit être utilisé dans l'instruction SQL.

12.2. Liste des commandes de périphériques (Events) queue `*_c'` table – structure

Cette table est une file d'événements/commandes pour chaque périphérique et a la structure suivante:

- `command` - commande à envoyer ou déjà envoyée au contrôleur
- `confirmed` - indicateur de confirmation s'il a déjà été envoyé et confirmé
- `date` - horodatage unix de l'événement
- `updated` - indicateur de mise à jour automatique (date/heure d'horodatage)

12.3. Accès aux résultats des bases de données - Niveau intermédiaire (lecture de données)

Les données peuvent être accessibles sans application Web frontale. Le système @City contient un script avec des fonctions de niveau intermédiaire. Les résultats sont renvoyés au format JSON.

12.3.1. Obtenir les statuts actuels de tous les appareils

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

La requête retourne la `'_ithings'` entière. (états actuels de tous les périphériques) au format 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": "73000200000f3600330268002400000002c002c002dfffffffffffffffff5b63000001c1000
001c2000000000000000009250a4f0a760a7a0a750a780a7e0000031d032205fc34029b025c02560046
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_kmh": "", "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. Obtenir des données historiques pour le périphérique

Interroger les données historiques d'un seul appareil par numero IMEI:

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

La table entière pouvant contenir des millions de lignes, elle doit être limitée à la clause WHERE afin de ne pas bloquer le serveur.

Paramètres supplémentaires d'url:

func – imeijson

imei – IMEI de l'appareil

field – champs à afficher dans les résultats (liste séparée par des virgules)

min – valeur minimale pour le premier champ de la liste

max – valeur maximale pour le premier champ de la liste

sort – champ de tri

tm – Ce champ est automatiquement ajouté aux résultats.

where – clause “où” pour limiter les données

Exemple:

Nous voulons obtenir le résultat suivant

pour appareil avec imei = 356345080018095

affiche les champs: ain5, ain6, gps_lat, gps_long

et limite ain5 dans la plage (1, 10000) - doit être le premier champ de la liste

et gps a des données valides (gps_fix = 3)

et date/heure du 2019-02-14 23:00:19 au 2019-02-15 00:00:00

Chaîne d'URL construite:

[http://iot.intelligentny.org/IoT/que.php?](http://iot.intelligentny.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.intelligentny.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')

Résultats de la requête:

```
[{ "ain5":"66", "ain6":"68", "gps_lat":"5202.7326N", "gps_long":"02115.8
073E", "tm":"2019-02-14 23:04:31" },
{ "ain5":"67", "ain6":"76", "gps_lat":"5202.7328N", "gps_long":"02115.80
75E", "tm":"2019-02-14 23:05:42" },
{ "ain5":"63", "ain6":"77", "gps_lat":"5202.7328N", "gps_long":"02115.80
74E", "tm":"2019-02-14 23:06:05" },
{ "ain5":"58", "ain6":"77", "gps_lat":"5202.7328N", "gps_long":"02115.80
75E", "tm":"2019-02-14 23:06:32" },
{ "ain5":"58", "ain6":"68", "gps_lat":"5202.7328N", "gps_long":"02115.80
```

```
76E", "tm": "2019-02-14 23:06:55" }]
```

12.3.3. Obtenir la liste des périphériques - un seul champ à partir des états actuels avec limitation

Cette fonction renvoie des données limitées de la table '_ithings'

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

Parameters:

func – fieldjson

field – champ à afficher dans les résultats - imei et tm sont automatiquement ajoutés

min - valeur minimale pour le champ

max - valeur maximale pour le champ

Pour la chaîne de requête ci-dessus, les résultats des champs ain5, imei, tm sont renvoyés:
si ain5 est dans la plage (13.5000)

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" } ]
```