

## I. Протокол передачі/прийому даних від облікової системи ver 8.1.0

### I.1. Принцип роботи AGB сервера.

AGB сервер працює в режимі асинхронної обробки подій (async event handler) - для якого основною задачею є прийом даних в режимі реального часу від кінцевого обладнання та, в разі потреби, моментальна передача цих даних до облікової системи. Це покладає на облікову систему певні вимоги по організації взаємодії з AGB сервером. Облікова система повинна забезпечити можливість двонаправленої передачі даних, при якій ініціаторами передачі виступають як облікова система, так і AGB сервер.

### I.2. Загальна характеристика протоколу.

Протокол призначений для обміну інформацією між AGB сервером та обліковою системою. AGB сервер підтримує взаємодію з обліковою системою через tcp-socket або через http запити. Прийнятний метод комунікації – через tcp-socket.

#### I.2.1 Ініціатори передачі даних.

Ініціатором передачі даних по мережевому з'єднанню може бути як AGB сервер, так і облікова система.

Якщо ініціатором передачі даних виступає облікова система то, спілкування між клієнтом і сервером відбувається по принципу запит-відповідь. На запити клієнта, AGB сервер завжди дає відповідь. В залежності від запиту – це можуть бути певні запитані дані, чи пакет статусу, який інформує про успішність виконання клієнтського запиту.

Якщо ініціатором передачі даних виступає AGB сервер – відповідь клієнта на отримання таких даних не потрібна. AGB сервер ініціює передачу даних тільки у випадках, коли облікова система не може знати про їх існування, але потребує їх.

#### I.2.2. Комунікація через tcp-socket.

Комунікація з AGB сервером для передачі/отримання інформації обліковою системою реалізована на основі стандартного мережевого tcp сокета через TLS захищений канал. AGB сервер виступає мережевим сервером, облікова система – мережевим клієнтом. Ініціація, створення та підтримка мережевого з'єднання в активному стані повністю покладається на клієнта(облікову систему). Стандартний TCP порт – 7354.

Дані, між сервером і клієнтом, передаються в формі пакетів даних. Пакет даних – це будь-які дані закодовані в base64 кодуванні. Ознака кінця пакету – символ переходу рядка - \n(0x0a).

**Приклад пакету даних:**

Інформаційний пакет:	{ "status": "Connect Ok" }
Пакет даних для передачі:	eyAic3RhdHVzIjogIkNvbW5lY3QgT2siIH0=\n

Інформаційний пакет для передачі представляється в json форматі в utf8 кодуванні.

**Приклади інформаційного пакету:**

- 1) { "status": "Connect Ok" }
- 2) json структура TTN:Create

#### I.2.3. Комунікація через http/https.

Враховуючи неможливість передачі даних по http протоколу в двонаправленому режимі, спілкування AGB сервера з обліковою системою реалізується в два етапи:

##### 1) Передача інформації від облікової системи до AGB сервера.

Для передачі інформації від облікової системи до AGB сервера, облікова система повинна

відправити http запит методом POST в JSON форматі в UTF8 кодуванні. В даному випадку AGB сервер виступає мережевим сервером, а облікова система – мережевим клієнтом. Пакет відправляє облікова система. Стандартний порт AGB сервера для http з'єднання – 7380, для https через tls захищений канал - 7381. Обов'язкова присутність поля **Content-Length** в заголовку HTTP пакету. Інші поля http заголовку не обробляються. У відповідь на клієнтський запит AGB сервер завжди відповідає HTTP кодом **200**, деталізація відповіді (пакет статусу, або запитані дані) міститься в json пакеті відповіді.

Для даного типу з'єднання інформаційні пакети, для виконання процедури авторизації, повинні доповнюватись двома полями:

```
"login": "master",
"password": "master"
```

або, здійснюватись авторизація стандартним методом HTTP base authorization.

## 2) Передача інформації від AGB сервера до облікової системи.

Для передачі інформації від AGB сервера до облікової системи AGB сервер відправляє http запит методом POST в JSON форматі в UTF8 кодуванні на сервер облікової системи. В даному випадку AGB сервер виступає мережевим клієнтом, а облікова система – мережевим сервером. Пакет відправляє AGB сервер. URL відправки пакетів задається в налаштуваннях AGB сервера. Для всіх інформаційних пакетів url однаковий, наприклад [http\(s\)://your\\_ip:your\\_port/agb](http(s)://your_ip:your_port/agb). Відповідь на пакет, отриманий від AGB сервера, облікова система повинна дати негайно, як тільки його отримає. Статус пакету завжди `Http.Status=200`. І тільки після відповіді на отриманий пакет, обробляти його згідно власних алгоритмів облікової системи. Логіка обробки пакетів на стороні облікової системи запускається в залежності від поля `procname` інформаційного пакету.

### I.2.4. Рекомендації щодо вибору протоколу спілкування.

Прийнятним до використання є метод взаємодії через tcp сокет. Взаємодія через http не рекомендована по причині неможливості обліковою системою зафіксувати та правильно відреагувати на момент неможливості передачі даних від AGB сервера до облікової системи. Вибір http протоколу означає, що конфігурація системи на поточному об'єкті дозволяє успішно функціонувати обліковій системі без отримання даних ініціатором яких виступає AGB сервер.

### I.2.5. Повторна відправка пакетів до облікової системи.

Якщо, при відправці пакету по http протоколу, AGB сервер не отримає відповіді від облікової системи на протязі 3 секунд, або відповідь не буде містити `http code 200`, пакет зберігається в FIFO стек для повторної передачі. Відправка пакетів зі стеку відбувається кожні 60 секунд з інтервалом відправки між пакетами 20 мілісекунд. Можливість повторної відправки задається примусово при конфігурації сервера. За замовчуванням, повторна відправка не відбувається. При зупинці сервера, стек повторної передачі пакетів зберігається.

### I.2.6. Пакет відповіді (стандартний пакет статусу).

AGB сервер завжди відповідає стандартним пакетом відповіді наступної структури:

Типи пакетів відповіді:

1) Інформація успішно оброблена:

```
{
  "procname": "найменування_запиту",
  "status": "1"
}
```

2) Інформація не оброблена (помилка):

```
{
  "procname": "найменування_запиту",
  "status": "-1",
  "status_name": "Опис помилки при обробці запиту"
}
```

Якщо, для якогось запиту пакет відповіді відрізняється від стандартного, то така відмінність буде зафіксована в описі конкретного запиту.

## II. Інформаційні пакети.

Допустима відсутність будь-якого поля в будь-якому пакеті якщо поле пакету не несе в собі ніякого смислового навантаження. Допустима присутність будь-якого поля в будь-якому пакеті, яке вважається службовим для AGB, та не потребує ніякої обробки та інтерпретації обліковою системою. Такі поля не описуються в цьому документі.

### II.1. Пакет AUTH – авторизація.

Після успішного з'єднання клієнта і сервера, першим пакетом який повинен відправити tcp-клієнт на AGB сервер є пакет авторизації, при використанні комунікації через http запити, даний пакет не відправляється.

JSON пакет AUTH має таку структуру:

```
{
  "login":          "master",
  "password":       "master",
  "procname":       "AUTH"
}
```

Опис полів:

login	Логін користувача зареєстрованого в системі AGB сервера
password	Пароль користувача зареєстрованого в системі AGB сервера
procname	Найменування функції обробки результату запиту. В даному випадку — AUTH

**При відправці пакету частіше одного разу на хвилину, пакет не обробляється!**

### II.2. Пакет TTN:Create – створення товаро-транспортної накладної.

Даний пакет відправляється при створенні документу. Якщо документ створюється в обліковій системі, пакет відправляється від облікової системи до AGB, інакше пакет відправляється від AGB до облікової системи.

JSON пакет даних запиту TTN:Create має таку структуру:

```
{
  "procname":          "TTN:Create",
  "control_point_num": "1",
  "status":            "1",
  "datetime":          "23.02.2016 08:21:01",
  "ttn_num":           "123",
  "org_id":            "1",
  "org_name":          "Найменування організації",
  "kontra_id":         "1",
  "kontra_name":       "Найменування контрагента",
  "driver_id":         "1",
  "driver_name":       "ФІО водія",
  "car_number":        "VI5678AT",
  "trailer_number":    "VI7896AT",
  "nomenclature_id":   "1",
  "nomenclature_name": "Найменування номенклатури",
  "carrier_id":        "1",
  "carrier_name":      "Найменування перевізника",
  "customer_id":       "1",
  "customer_name":     "Найменування замовника",
  "sender_id":         "1",
  "sender_name":       "Найменування відправника",
  "loadingpoint_id":   "1",
  "loadingpoint_name": "Найменування пункту завантаження",
  "cardnumber":        "номер картки",
  "transport_type":    "тип транспортного засобу",
  "separate_weighing": "0",
  "tipping_method":    "Метод вивантаження",
  "in_ttn_num":        "ad589",
  "in_ttn_brutto":     "3100",
  "in_ttn_netto":      "1000",
  "in_ttn_tara":       "2100",
}
```

```
"sample":      "1"
}
```

**Опис полів:**

procname	Найменування функції обробки результату запиту. В даному випадку — <b>TTN:Create</b>
control_point_num	Унікальний номер контрольної точки (в нашому випадку місця ресстрації) (згідно бази даних AGB сервера). (Поле num пакету Controlpoint)
status	Статус (Поле num пакету Logistic): <b>1 - вивантаження.</b> Якщо для процесу вивантаження існує кілька логістичних правил проходження контрольних точок, то поле status повинно приймати унікальні <u>додатні значення</u> (наприклад: 2, 3, тощо ...) <b>-1 - завантаження.</b> Якщо для процесу завантаження існує кілька логістичних правил проходження контрольних точок, то поле status повинно приймати унікальні <u>від'ємні значення</u> (Наприклад: -2, -3, тощо ...)
datetime	Дата та час створення товаро-транспортної накладної в форматі «dd.mm.yyyy hh:mm:ss»
ttn_num	Унікальний номер товаро-транспортної накладної
org_id	Унікальний ідентифікатор організації (елеватор, пункт зберігання, тощо ...) згідно довідника облікової системи. Служить для синхронізації довідників AGB та облікової системи. При відсутності довідника в обліковій системі значення ідентифікатора дорівнює нулю.
org_name	Найменування організації
kontra_id	Унікальний ідентифікатор контрагента згідно довідника облікової системи. Служить для синхронізації довідників AGB та облікової системи. При відсутності довідника в обліковій системі значення ідентифікатора дорівнює нулю.
kontra_name	Найменування контрагента.
driver_id	Унікальний ідентифікатор водія згідно довідника облікової системи. Служить для синхронізації довідників AGB та облікової системи. При відсутності довідника в обліковій системі значення ідентифікатора дорівнює нулю.
driver_name	ФІО водія
car_number	Державний номер автомобіля при transport_type = 1, та номер вагону при transport_type = 2
trailer_number	Державний номер причепа
nomenclature_id	Унікальний ідентифікатор номенклатури згідно довідника облікової системи. Служить для синхронізації довідників AGB та облікової системи. При відсутності довідника в обліковій системі значення ідентифікатора дорівнює нулю.
nomenclature_name	Найменування номенклатури
carrier_id	Унікальний ідентифікатор перевізника згідно довідника облікової системи. Служить для синхронізації довідників AGB та облікової системи. При відсутності довідника в обліковій системі значення ідентифікатора дорівнює нулю.
carrier_name	Найменування перевізника
customer_id	Унікальний ідентифікатор замовника згідно довідника облікової системи. Служить для синхронізації довідників AGB та облікової системи. При відсутності довідника в обліковій системі значення ідентифікатора дорівнює нулю.
customer_name	Найменування замовника
sender_id	Унікальний ідентифікатор відправника згідно довідника облікової системи. Служить для синхронізації довідників AGB та облікової системи. При відсутності довідника в обліковій системі значення ідентифікатора дорівнює нулю.
sender_name	Найменування відправника
loadingpoint_id	Унікальний ідентифікатор пункту завантаження згідно довідника облікової системи. Служить для синхронізації довідників AGB та облікової системи. При відсутності довідника в обліковій системі значення ідентифікатора дорівнює нулю.
loadingpoint_name	Найменування пункту завантаження
cardnumber	Номер ідентифікаційної картки (Ціле додатнє значення)
transport_type	Тип транспортного засобу: <b>1</b> - автомобільний транспорт <b>2</b> - залізничний транспорт
separate_weighing	Тип зважування Приклади: 0 - коли авто з причепом зважується за один раз (сумісно). В накладній присутні і номер авто і номер

	причепа 1 - якщо зважується авто без причепа (в накладній тільки номер авто) 1 - також стосується зважування вагона (в накладній присутній номер вагону) 2 - це коли зважується причеп роздільно від авто (після попереднього зважування авто) в накладній присутній номер авто і номер причепа
tipping_method	Метод вивантаження, може приймати одне із наступних значень: lifert - підйомник tipper - самоскид будь-яке інше значення - метод вивантаження не визначений
in_ttn_num	Номер вхідної ТТН
in_ttn_brutto	Вага брутто згідно вхідної ТТН
in_ttn_netto	Вага нетто згідно вхідної ТТН
in_ttn_tara	Вага тари згідно вхідної ТТН

### II.3. Пакет TTNWeighing:Create – зважування.

Даний пакет відправляється при створенні документу зважування. Якщо документ створюється в обліковій системі, пакет відправляється від облікової системи до AGB, інакше пакет відправляється від AGB до облікової системи.

JSON пакет даних запиту TTNWeighing:Create має таку структуру:

```
{
  "procname":      "TTNWeighing:Create",
  "control_point_num": "2",
  "status":        "1",
  "datetime":      "23.02.2016 08:21:01",
  "ttn_num":       "123",
  "weight":        "60000",
  "separate_weighing": "1",
  "deltaClockseconds": "50",
  "clockseconds":  "1578388319",
  "driverisaccept": "1"
}
```

#### Опис полів:

procname	Найменування функції обробки результату запиту. В даному випадку — <b>TTNWeighing:Create</b>
control_point_num	Унікальний номер контрольної точки (в нашому випадку вагової) (згідно бази даних AGB сервера). (Поле num пакету Controlpoint)
status	Статус (вага брутто/вага тари): 1 — вага брутто -1 — вага тари
datetime	Дата та час створення документу зважування в форматі «dd.mm.yyyy hh:mm:ss»
ttn_num	Номер товаро-транспортної накладної
weight	Покази ваг
separate_weighing	Номер роздільного зважування. 1 - (авто), 2 - (причеп 1), 3 - (причеп 2) . . . N - (причеп N-1), 0 - сумісне(одночасне) зважування (авто+причеп) Приклади: 0 - коли авто з причепом зважується за один раз (сумісно). В накладній присутні і номер авто і номер причепа 1 - якщо зважується авто без причепа (в накладній тільки номер авто) або зважується саме авто роздільно від причепа, хоча є причеп (в накладній задано номер авто і номер причепа), 1 - також стосується зважування вагона (в накладній присутній номер вагону) 2 - це коли зважується причеп роздільно від авто (після попереднього зважування авто) в накладній присутній номер авто і номер причепа
deltaClockseconds	Різниця часу, виражена в секундах (ціле додатне значення), між моментом відправки пакету та моментом останнього поточного зважування, та визначається за формулою: $deltaClockseconds = t2 - t1$ де, t2 — момент часу передачі пакету (натискання кнопки «Зважити та провести (створити документ)») t1 — момент часу останнього поточного зважування (останнє натискання кнопки «Зважити»)
clockseconds	Момент часу підтвердження ваги водієм при автоматичному зважуванні, виражений в секундах від

	початку епохи. (Unix time)
driverisaccept	1 — водій погодився з вагою, 0 — водій не погодився з вагою

У відповідь на запит TTNWeighing:Create AGB сервер завжди відповідає пакетом статусу:

#### Типи пакетів відповіді:

##### 1) Інформація успішно оброблена:

```
{
  "procname":      "TTNWeighing:Create",
  "status": "1",
  "inbox": "1",
  "inbox_datetime": "01.01.1970 00:00:01"
}
```

Де:

**inbox** може приймати наступні значення:

1 – порушення периметру відсутнє

0 – контроль периметру не підтримується для даної контрольної точки

-1 – зафіксовано порушення периметру

**inbox\_datetime** - дата та час спрацювання події, при inbox=0 значення поля відсутнє

##### 2) Інформація не оброблена (помилка):

```
{
  "procname":      "TTNWeighing:Create",
  "status":      "-1",
  "status_name": "Опис помилки при обробці запиту"
}
```

## II.4. Пакет TTNLab:Create – дані лабораторного аналізу.

Даний пакет відправляється при створенні документу. Якщо документ створюється в обліковій системі, пакет відправляється від облікової системи до AGB, інакше пакет відправляється від AGB до облікової системи.

JSON пакет даних запиту TTNLab:Create має таку структуру:

```
{
  "procname":      "TTNLab:Create",
  "control_point_num": "2",
  "datetime":      "23.02.2016 08:21:01",
  "ttn_num":       "123",
  "unloadingpoint_id": "1",
  "unloadingpoint_name": "Найменування пункту вивантаження",
  "unloadingpoint_num": "Номер на табло",
  "storageplace_id": "1",
  "storageplace_name": "Найменування Місця зберігання",
  "nomenclature_id": "1",
  "nomenclature_name": "Найменування номенклатури",
  "analysis": [{
    "parameter_name": "вологість",
    "parameter_value": "25",
    "parameter_unit": "%"
  }, {
    "parameter_name": "засміченість",
    "parameter_value": "10",
    "parameter_unit": "%"
  }, {
    "parameter_name": "інфікованість",
    "parameter_value": "відсутня",
    "parameter_unit": ""
  }
  ],
  "isaccept": "1",
  "class":    "клас продукції",
  "provenance": "походження продукції",
  "sample":   "1"
}
```

**Опис полів:**

procname	Найменування функції обробки результату запиту. В даному випадку — <b>TTNLab:Create</b>	
control_point_num	Унікальний номер контрольної точки (в нашому випадку Лабораторії) (згідно бази даних AGB сервера). (Поле num пакету Controlpoint)	
datetime	Дата та час створення документу лабораторного аналізу в форматі «dd.mm.yyyu hh:mm:ss»	
ttn_num	Номер товаро-транспортної накладної	
unloadingpoint_id	Унікальний ідентифікатор пункту вивантаження згідно довідника облікової системи. Служить для синхронізації довідників AGB та облікової системи. При відсутності довідника в обліковій системі значення ідентифікатора дорівнює нулю.	
unloadingpoint_name	Найменування пункту вивантаження	
unloadingpoint_num	Номер на табло(ціле додатне значення)	
storageplace_id	Унікальний ідентифікатор місця зберігання згідно довідника облікової системи. Служить для синхронізації довідників AGB та облікової системи. При відсутності довідника в обліковій системі значення ідентифікатора дорівнює нулю.	
storageplace_name	Найменування місця зберігання	
isaccept	Відповідність аналізу певним нормам: 1 — аналіз прийнятний, 0 — аналіз неприйнятний	
nomenclature_id	Унікальний ідентифікатор номенклатури згідно довідника облікової системи. Служить для синхронізації довідників AGB та облікової системи. При відсутності довідника в обліковій системі значення ідентифікатора дорівнює нулю.	
nomenclature_name	Найменування номенклатури	
analysis	Лабораторний аналіз. Масив значень в залежності від кількості проведених лабораторних аналізів.	
	parameter_name	Найменування параметру проведеного аналізу
	parameter_value	Значення параметру проведеного аналізу
	parameter_unit	Одиниця виміру параметру проведеного аналізу
class	Клас продукції	
provenance	Походження продукції	
sample	Номер проби	

**II.5. Пакет TTN:LogisticChange – зміна логістичного правила по накладній.**

Даний пакет відправляється при зміні логістичного правила по накладній. Якщо така зміна відбулась в обліковій системі, пакет відправляється від облікової системи до AGB, інакше пакет відправляється від AGB до облікової системи.

JSON пакет даних запиту TTN:LogisticChange має таку структуру:

```
{
  "procname":      "TTN:LogisticChange",
  "ttn_num":       "20191001132003",
  "logistic_num":  "2"
}
```

**Опис полів:**

procname	Найменування функції обробки результату запиту. В даному випадку — <b>TTN:LogisticChange</b>	
ttn_num	Номер товаро-транспортної накладної	
logistic_num	Номер нового логістичного правила (згідно бази даних AGB сервера) (Поле num пакету Logistic)	

**II.6. Пакет EventCard - спрацювання карткозчитувача.**

Передача пакета AGB сервером відбудеться при спрацюванні карткозчитувача на будь-якій з контрольних точок з встановленим карткозчитувачем.

JSON пакет даних EventCard має таку структуру:

```
{
```

```

    "procname":      "EventCard",
    "cardnumber":   "3825698",
    "control_point_num": "2",
    "controller_id": "56975",
    "clockseconds": "1578388319"
}

```

**Опис полів:**

procname	Найменування функції. В даному випадку — <b>EventCard</b>
cardnumber	Номер ідентифікаційної картки
control_point_num	Унікальний номер контрольної точки (згідно бази даних AGB сервера) на якій відбулося спрацювання карткозчитувача (Поле num пакету Controlpoint)
controller_id	Унікальний ідентифікатор контроллера карткозчитувача (згідно бази даних AGB сервера) (Поле controller_id пакету Sensor)
clockseconds	Момент часу виникнення події, виражений в секундах від початку епохи. (Unix time)

**II.7. Пакет EventSensor – спрацювання сенсора.**

Передача пакета AGB сервером відбудеться при спрацюванні сенсора(оптопара, ...) на будь-якій з контрольних точок з встановленим відповідним обладнанням.

JSON пакет даних EventSensor має таку структуру:

```

{
  "procname":      "EventSensor",
  "control_point_num": "3",
  "controller_id": "67823",
  "chan":          "01",
  "sensorval":    "1",
  "clockseconds": "1578388319"
}

```

**Опис полів:**

procname	Найменування функції. В даному випадку — <b>EventSensor</b>
control_point_num	Унікальний номер контрольної точки (згідно бази даних AGB сервера) на якій відбулося спрацювання сенсора (Поле num пакету Controlpoint)
controller_id	Унікальний ідентифікатор контроллера сенсора (згідно бази даних AGB сервера) (Поле controller_id пакету Sensor)
chan	Канал контроллера (Поле chan пакету Sensor)
sensorval	Значення каналу контроллера
clockseconds	Момент часу виникнення події, виражений в секундах від початку епохи. (Unix time)

**II.8. Пакет EventRecognit – дані по розпізнаванню державного номерного знаку.**

Передача пакета AGB сервером відбудеться при отриманні даних від системи розпізнавання державного номерного знаку на будь-якій з контрольних точок з встановленим відповідним обладнанням, та при відповідному налаштуванні автоматичної передачі даних обліковій системі в AGB.

JSON пакет даних EventRecognit має таку структуру:

```

{
  "procname":      "EventRecognit",
  "control_point_num": "4",
  "camera_chan":   "JbSdA",
  "events_id":     "1259876",
  "plate":         "BI6754AH",
  "quality":       "78.77",
  "UTC_time_bestview": "1573827687926319"
}

```



**Опис полів:**

procname	Найменування функції. В даному випадку — <b>EventRecognit</b>
control_point_num	Унікальний номер контрольної точки (згідно бази даних AGB сервера) на якій відбулося спрацювання сенсора (Поле num пакету Controlpoint)
camera_chan	Унікальний ідентифікатор камери, підключеної до системи розпізнавання (Поле chan пакету Camera)
events_id	Унікальний ідентифікатор події (згідно бази даних системи розпізнавання)
plate	Розпізнаний державний номерний знак
quality	Ймовірність розпізнавання виражена в відсотках
UTC_time_bestview	UTC час найкращого розпізнавання виражена в мікросекундах

**II.9. Пакет Logistic – логістичне правило.**

Передача пакета AGB сервером відбувається при зміні чи створенні логістичного правила в довіднику логістичних правил.

JSON інформаційний пакет Logistic має таку структуру:

```
{
  "procname":      "Logistic",
  "num":           "1",
  "name":          "Вивантаження через вагову 70 тон"
}
```

**Опис полів:**

procname	Найменування функції. В даному випадку — <b>Logistic</b>
num	Унікальний номер логістичного правила згідно бази даних AGB
name	Найменування логістичного правила

**II.10. Пакет Controlpoint – контрольна точка.**

Передача пакета AGB сервером відбувається при зміні чи створенні контрольної точки в довіднику контрольних точок.

JSON інформаційний пакет Controlpoint має таку структуру:

```
{
  "procname":      "Controlpoint",
  "num":           "1",
  "name":          "Вагова 70 тон"
}
```

**Опис полів:**

procname	Найменування функції. В даному випадку — <b>Controlpoint</b>
num	Унікальний номер контрольної точки згідно бази даних AGB
name	Найменування контрольної точки

**II.11. Пакет Camera – камера.**

Передача пакета AGB сервером відбувається при зміні чи створенні камери в довіднику камер.

JSON інформаційний пакет Camera має таку структуру:

```
{
  "procname":      "Camera",
  "control_point_num": "1",
  "chan":          "JbSdA",
  "name":          "Вагова 70 тон (Кузов)"
}
```

}

**Опис полів:**

procname	Найменування функції. В даному випадку — <b>Camera</b>
control_point_num	Унікальний номер контрольної точки (згідно бази даних AGB сервера) до якої прив'язана камера
chan	Унікальний ідентифікатор камери згідно системи відеоспостереження
name	Найменування камери

**II.12. Пакет Sensor – сенсор.**

Передача пакета AGB сервером відбувається при зміні чи створенні сенсора в довіднику сенсорів.

JSON інформаційний пакет Sensor має таку структуру:

```
{
  "procname":      "Sensor",
  "control_point_num": "1",
  "controller_id":  "896745",
  "chan":          "JbSdA",
  "name":          "Оптопара на в`їзді на вагову 70 тон"
}
```

**Опис полів:**

procname	Найменування функції. В даному випадку — <b>Sensor</b>
control_point_num	Унікальний номер контрольної точки (згідно бази даних AGB сервера) до якої прив'язаний сенсор або карткозчитувач
controller_id	Унікальний ідентифікатор контроллера до якого підключений сенсор або карткозчитувач
chan	Унікальний ідентифікатор каналу сенсора
name	Найменування сенсора або карткозчитувача

**II.13. Пакет TTN:SeparateWeighingChange – зміна типу зважування по накладній.**

Даний пакет відправляється при зміні типу зважування по накладній (Поле separate\_weighing пакету TTN:Create). Якщо така зміна відбулась в обліковій системі, пакет відправляється від облікової системи до AGB, інакше пакет відправляється від AGB до облікової системи.

JSON пакет даних запиту TTN:SeparateWeighingChange має таку структуру:

```
{
  "procname":      "TTN:SeparateWeighingChange",
  "ttn_num":       "20191001132003",
  "separate_weighing": "2"
}
```

**Опис полів:**

procname	Найменування функції обробки результату запиту. В даному випадку — <b>TTN:SeparateWeighingChange</b>
ttn_num	Номер товаро-транспортної накладної
separate_weighing	Тип зважування (Поле separate_weighing пакету TTN:Create)

**II.14. Пакет TTNLab:IsAcceptChange – зміна прийнятності лабораторного аналізу.**

Даний пакет відправляється при зміні прийнятності лабораторного аналізу по накладній

(Поле isaccept пакету TTNLab:Create). Якщо така зміна відбулась в обліковій системі, пакет відправляється від облікової системи до AGB, інакше пакет відправляється від AGB до облікової системи.

JSON пакет даних запиту TTNLab:IsAcceptChange має таку структуру:

```
{
  "procname":      "TTNLab:IsAcceptChange",
  "ttn_num":       "20191001132003",
  "isaccept":      "0"
}
```

#### Опис полів:

procname	Найменування функції обробки результату запиту. В даному випадку — <b>TTNLab:IsAcceptChange</b>
ttn_num	Номер товаро-транспортної накладної
isaccept	Відповідність аналізу певним нормам: 1 — аналіз прийнятний, 0 — аналіз неприйнятний (Поле isaccept пакету TTNLab:Create)

### II.15. Пакет TTN:Forcibly – примусове закриття накладної.

Даний пакет відправляється при примусовому закритті накладної. Якщо закриття відбулося в обліковій системі, пакет відправляється від облікової системи до AGB, інакше пакет відправляється від AGB до облікової системи.

JSON пакет даних запиту TTN:Forcibly має таку структуру:

```
{
  "procname":      "TTN:Forcibly",
  "ttn_num":       "20191001132003",
  "comment":       "Технічна несправність автомобіля унеможлиблює продовження вивантаження"
}
```

#### Опис полів:

procname	Найменування функції обробки результату запиту. В даному випадку — <b>TTN:Forcibly</b>
ttn_num	Номер товаро-транспортної накладної
comment	Описовий текст причини примусового закриття накладної

### II.16. Пакет TTNLab:UnloadingpointChange – зміна пункту вивантаження.

Даний пакет відправляється при зміні пункту вивантаження (Поля unloadingpoint\_id, unloadingpoint\_name, unloadingpoint\_num пакету TTNLab:Create). Якщо така зміна відбулась в обліковій системі, пакет відправляється від облікової системи до AGB, інакше пакет відправляється від AGB до облікової системи.

JSON пакет даних запиту TTNLab:UnloadingpointChange має таку структуру:

```
{
  "procname":      "TTNLab:UnloadingpointChange",
  "ttn_num":       "20191001132003",
  "unloadingpoint_id": "1",
  "unloadingpoint_name": "Найменування пункту вивантаження",
  "unloadingpoint_num": "Номер на табло",
}
```

#### Опис полів:

procname	Найменування функції обробки результату запиту. В даному випадку — <b>TTNLab:UnloadingpointChange</b>
ttn_num	Номер товаро-транспортної накладної
unloadingpoint_id	Унікальний ідентифікатор пункту вивантаження згідно довідника облікової системи. Служить для синхронізації довідників AGB та облікової системи. При відсутності довідника в обліковій системі значення ідентифікатора дорівнює нулю.
unloadingpoint_name	Найменування пункту вивантаження

unloadingpoint_num	Номер на табло(ціле додатне значення)
--------------------	---------------------------------------

## II.17. Пакет TTN:CardnumChange – зміна ID Картки активної накладної в результаті втрати поточної картки.

Даний пакет відправляється при зміні ID Картки активної накладної в результаті втрати поточної картки по накладній. Якщо зміна відбулась в обліковій системі, пакет відправляється від облікової системи до AGB, інакше пакет відправляється від AGB до облікової системи.

JSON пакет даних запиту TTN:CardnumChange має таку структуру:

```
{
  "procname":      "TTN:CardnumChange",
  "ttn_num":       "20191001132003",
  "cardnumber":    "Номер нової ID Картки"
}
```

### Опис полів:

procname	Найменування функції обробки результату запиту. В даному випадку — <b>TTN:CardnumChange</b>
ttn_num	Номер товаро-транспортної накладної
cardnumber	Номер нової ID Картки

## II.18. Пакет TTNLab:StorageplaceChange – зміна місця зберігання.

Даний пакет відправляється при зміні місця зберігання (Поля storageplace\_id, storageplace\_name пакету TTNLab:Create). Якщо така зміна відбулась в обліковій системі, пакет відправляється від облікової системи до AGB, інакше пакет відправляється від AGB до облікової системи.

JSON пакет даних запиту TTNLab:StorageplaceChange має таку структуру:

```
{
  "procname":      "TTNLab:StorageplaceChange",
  "ttn_num":       "20191001132003",
  "storageplace_id": "1",
  "storageplace_name": "Найменування місця зберігання"
}
```

### Опис полів:

procname	Найменування функції обробки результату запиту. В даному випадку — <b>TTNLab:StorageplaceChange</b>
ttn_num	Номер товаро-транспортної накладної
storageplace_id	Унікальний ідентифікатор місця зберігання згідно довідника облікової системи. Служить для синхронізації довідників AGB та облікової системи. При відсутності довідника в обліковій системі значення ідентифікатора дорівнює нулю.
storageplace_name	Найменування місця зберігання

## II.19. Пакет Driver – водій.

Передача пакета AGB сервером відбувається при зміні чи створенні водія в довіднику водіїв.

JSON пакет даних запиту Driver має таку структуру:

```
{
  "procname":      "Driver",
  "id":            "21",
  "pid":          "2g6143945-5623je0",
  "name":         "Іванов Іван Іванович",
  "phone":        "0505555050 0676776767"
}
```

### Опис полів:

procname	Найменування функції обробки результату запиту. В даному випадку — <b>Driver</b>
id	Унікальний ідентифікатор водія в базі даних AGB
pid	Унікальний ідентифікатор водія в базі даних облікової системи. Якщо об'єкт створювався в AGB, то id=pid
name	Прізвище, ім'я, по-батькові водія
phone	Список з одного або більше слів розділених пробілами. Одне слово — один номер телефону

### III. Запити.

#### III.1. Запит EventRecognit:Get – отримання даних по розпізнаванню.

Запит призначений для отримання даних по розпізнаванню державного номера за певний проміжок часу. Запит відправляється тільки від облікової системи до AGB сервера.

JSON пакет даних EventRecognit:Get має таку структуру:

```
{
  "procname":      "EventRecognit:Get",
  "camera_chan":   "JbSdA",
  "clockseconds_begin": "1575639607",
  "clockseconds_end":  "1575639608",
  "control_point_num": "2"
}
```

##### Опис полів:

procname	Найменування функції. В даному випадку — <b>EventRecognit:Get</b>
camera_chan	Унікальний ідентифікатор камери, підключеної до системи розпізнавання (Поле chan пакету Camera)
clockseconds_begin	Початковий момент періоду за який необхідно отримати інформацію по розпізнаванню виражений в секундах від початку епохи. (Unix time)
clockseconds_end	Кінцевий момент періоду
control_point_num	Унікальний номер контрольної точки (згідно бази даних AGB сервера) до якої прив'язана камера (Поле num пакету Controlpoint)

Одновременна наявність обох полів camera\_chan та control\_point\_num необов'язкова. Присутність одного з них – обов'язкова. Якщо необхідно отримати інформацію по конкретній камері – задається поле camera\_chan, якщо по всіх камерах, розташованих на контрольній точці, – задається поле control\_point\_num.

##### Типи пакетів відповіді:

###### 1) Інформація успішно оброблена, є дані для передачі:

Пакети EventRecognit, по одному пакету даних на кожен інформаційний пакет. Для з'єднання через http інформаційний пакет додатково огортається в структуру array.

###### 2) Інформація успішно оброблена, дані відсутні:

```
{
  "procname":      "EventRecognit:Get",
  "status": "1"
}
```

###### 3) Інформація не оброблена (помилка):

```
{
  "procname":      "EventRecognit:Get",
  "status":        "-1",
  "status_name":   "Опис помилки при обробці запиту"
}
```

#### III.2. Запит Logistic:Get – список логістичних правил.

Запит призначений для отримання списку логістичних правил. Пакет відправляється тільки від облікової системи до AGB сервера.

JSON пакет запиту Logistic:Get має таку структуру:

```
{
  "procname":      "Logistic:Get"
}
```

#### Опис полів:

procname	Найменування функції. В даному випадку — <b>Logistic:Get</b>
----------	--

#### Типи пакетів відповіді:

##### 1) Інформація успішно оброблена, є дані для передачі:

Пакети Logistic, по одному пакету даних на кожен інформаційний пакет. Для з'єднання через http інформаційний пакет додатково огортається в структуру array.

##### 2) Інформація успішно оброблена, дані відсутні:

```
{
  "procname":      "Logistic:Get",
  "status": "1"
}
```

##### 3) Інформація не оброблена (помилка):

```
{
  "procname":      "Logistic:Get",
  "status":        "-1",
  "status_name":   "Опис помилки при обробці запиту"
}
```

**При відправці запиту частіше одного разу на хвилину, запит не обробляється!**

### III.3. Запит Controlpoint:Get – список контрольних точок.

Запит призначений для отримання списку контрольних точок. Пакет відправляється тільки від облікової системи до AGB сервера.

JSON пакет запиту Controlpoint:Get має таку структуру:

```
{
  "procname":      "Controlpoint:Get"
}
```

#### Опис полів:

procname	Найменування функції. В даному випадку — <b>Controlpoint:Get</b>
----------	--

#### Типи пакетів відповіді:

##### 1) Інформація успішно оброблена, є дані для передачі:

Пакети Controlpoint, по одному пакету даних на кожен інформаційний пакет. Для з'єднання через http інформаційний пакет додатково огортається в структуру array.

##### 2) Інформація успішно оброблена, дані відсутні:

```
{
  "procname":      "Controlpoint:Get",
  "status": "1"
}
```

##### 3) Інформація не оброблена (помилка):

```
{
  "procname":      "Controlpoint:Get",
  "status":        "-1",
  "status_name":   "Опис помилки при обробці запиту"
}
```

**При відправці запиту частіше одного разу на хвилину, запит не обробляється!**

### III.4. Запит Camera:Get – список камер.

Запит призначений для отримання списку камер. Пакет відправляється тільки від

облікової системи до AGB сервера.

JSON пакет запиту Camera:Get має таку структуру:

```
{
  "procname":      "Camera:Get"
}
```

**Опис полів:**

procname	Найменування функції. В даному випадку — <b>Camera:Get</b>
----------	--

**Типи пакетів відповіді:**

**1) Інформація успішно оброблена, є дані для передачі:**

Пакети Camera, по одному пакету даних на кожен інформаційний пакет. Для з'єднання через http інформаційний пакет додатково огортається в структуру array.

**2) Інформація успішно оброблена, дані відсутні:**

```
{
  "procname":      "Camera:Get",
  "status": "1"
}
```

**3) Інформація не оброблена (помилка):**

```
{
  "procname":      "Camera:Get",
  "status":        "-1",
  "status_name":   "Опис помилки при обробці запиту"
}
```

**При відправці запиту частіше одного разу на хвилину, запит не обробляється!**

### III.5. Запит Sensor:Get – список сенсорів та карткозчитувачів.

Запит призначений для отримання списку сенсорів та карткозчитувачів. Пакет відправляється тільки від облікової системи до AGB сервера.

JSON пакет запиту Sensor:Get має таку структуру:

```
{
  "procname":      "Sensor:Get"
}
```

**Опис полів:**

procname	Найменування функції. В даному випадку — <b>Sensor:Get</b>
----------	--

**Типи пакетів відповіді:**

**1) Інформація успішно оброблена, є дані для передачі:**

Пакети Sensor, по одному пакету даних на кожен інформаційний пакет. Для з'єднання через http інформаційний пакет додатково огортається в структуру array.

**2) Інформація успішно оброблена, дані відсутні:**

```
{
  "procname":      "Sensor:Get",
  "status": "1"
}
```

**3) Інформація не оброблена (помилка):**

```
{
  "procname":      "Sensor:Get",
  "status":        "-1",
  "status_name":   "Опис помилки при обробці запиту"
}
```

**При відправці запиту частіше одного разу на хвилину, запит не обробляється!**

### III.6. Запит Tablo:SendMessage – відображення інформації на AGBTablo.

Запит призначений для візуалізації інформації на AGBTablo. Всі повідомлення ставляться в чергу FIFO та відображаються на табло на протязі певного інтервалу часу.

JSON пакет запиту Tablo:SendMessage має таку структуру:

```
{
  "procname":      "Tablo:SendMessage",
  "control_point_num": "1",
  "ttn_num":       "1",
  "message":       "Володимир Іванович\nІнформація успішно оброблена.",
  "color":         "\$:Main(cfg,messagetodo_color)"
}
```

**Опис полів:**

procname	Найменування функції. В даному випадку — <b>Tablo:SendMessage</b>
control_point_num	Унікальний номер контрольної точки (згідно бази даних AGB сервера) до якої прив'язане ARGTablo
ttn_num	Номер ТТН. Обов'язковий параметр при використанні MobiCard в режимі мобільного додатку водія
message	Повідомлення для відображення на табло. Для переходу на новий рядок можна використовувати двосимвольне мнемозначення \n
color	Foreground колір повідомлення(колір шрифту). Може бути пустим, для відображення кольором за замовчуванням(чорний), або приймати значення: \\$:Main(cfg,messagetodo_color) — повідомлення про наступну дію для водія \\$:Main(cfg,violation_color) — повідомлення для візуалізації порушення

### III.7. Запит Weight:ToZero – Обнулення вагопроцесора.

Запит призначений для обнулення вагопроцесора.

JSON пакет запиту Weight:ToZero має таку структуру:

```
{
  "procname":      "Weight:ToZero",
  "controller_id": "15698",
  "chan":          "W1"
}
```

**Опис полів:**

procname	Найменування функції. В даному випадку — <b>Weight:ToZero</b>
controller_id	Унікальний ідентифікатор контроллера згідно даних, представлених в пакеті Sensor запиту Sensor:Get
chan	Канал контроллера

### III.8. Запит Driver:Get – список водіїв.

Запит призначений для отримання списку водіїв. Пакет відправляється тільки від облікової системи до AGB сервера.

JSON пакет запиту Driver:Get має таку структуру:

```
{
  "procname":      "Driver:Get"
}
```

**Опис полів:**

procname	Найменування функції. В даному випадку — <b>Driver:Get</b>
----------	--

**Типи пакетів відповіді:**

**1) Інформація успішно оброблена, є дані для передачі:**

Пакети Driver, по одному пакету даних на кожен інформаційний пакет. Для з'єднання через http інформаційний пакет додатково огортається в структуру array.

**2) Інформація успішно оброблена, дані відсутні:**



```
{
  "procname":      "Driver:Get",
  "status": "1"
}

3) Інформація не оброблена (помилка):
{
  "procname":      "Driver:Get",
  "status":        "-1",
  "status_name":   "Опис помилки при обробці запиту"
}
```

При відправці запиту частіше одного разу на хвилину, запит не обробляється!

#### IV. Ідентифікатори, ключові та унікальні поля.

Загалом, реляційні бази даних, при операціях з об'єктами, оперують унікальними ключовими ідентифікаторами (primary key в термінології реляційних баз даних).

При формуванні запиту від облікової системи до AGB сервера в json пакеті відповідних полів вказуються primary key бази даних облікової системи. (Наприклад: поле kontra\_id запиту TTN:Create)

При створенні і передачі пакету даних, ініційованого AGB сервером, в json пакеті відповідних полів вказуються primary key бази даних AGB сервера.

Для AGB номер товаро-транспортної накладної є унікальним. Якщо облікова система мають інший підхід до унікальності ttn\_num, необхідно привести його до унікальності.

#### VI. Сторонні повідомлення облікової системи.

AGB сервер може зберігати та обробляти повідомлення облікової системи які не пов'язані з алгоритмами роботи AGB. Ці повідомлення можуть надходити від облікової системи в будь-який момент, та мати будь-який смисл. Повідомлення певним чином відображаються в AGB та, в момент отримання, можуть виконуватись певні дії.

##### VI.1. Пакет Message – відправка повідомлення.

Пакет призначений для відправки повідомлення від облікової системи до AGB сервера.

JSON пакет даних Message має таку структуру:

```
{
  "procname":      "Message",
  "message_id":    "17659846",
  "type":          "1",
  "name":          "Проведення лабораторного аналізу затримується",
  "floatname":     "до 14:00 годин",
  "ttn_num":       "номер товаро-транспортної накладної",
  "controlpoint_num": "Номер контрольної точки",
  "isphoto":       "0"
}
```

Опис полів:

procname	Найменування функції. В даному випадку — <b>Message</b>
message_id	Унікальний ідентифікатор повідомлення згідно довідника облікової системи. Служить для синхронізації довідників AGB та облікової системи. При відсутності довідника в обліковій системі значення ідентифікатора дорівнює нулю.
type	Тип повідомлення: 1 - інформаційне повідомленнями -1 - порушення
name	Групове найменування повідомлення (зберігається в довіднику)
floatname	Додаткове смислове уточнення до групового найменування повідомлення (має суто інформаційний характер)
ttn_num	Номер товаро-транспортної накладної (необов'язкове поле)

controlpoint_num	Номер контрольної точки (обов'язкове, якщо задане поле ttn_num)
isphoto	Зфотографувати всіма камерами розміщеними на контрольній точці (необов'язкове поле, значення за замовчуванням - 0)