

Довідник PoolLab2 WSAPI

Water-i.d. GmbH

Зміст

1 Вступ	2
2 WiFi Інтерфейс	3
2.1 Взаємодія з користувачем	3
3 Вимоги до WiFi та мережі	4
3.1 Version and Security	4
3.2 (B)SSID	4
3.3 IP / DHCP	4
3.4 NTP	4
4 Вимоги до серверу / криптографії	5
4.1 Socket	5
4.2 SSL	5
4.3 Websocket	5
5 Конфігурація	6
5.1 WiFi Логіни	6
5.2 Оновлення шляху API	6
5.3 Шлях до хмарного API	6
5.4 Логіни хмарного API	6
6 WiFi-процес	7
7 Кінцеві точки API	8
7.1 Кінцева точка API оновлення	8
7.1.1 Запит кінцевої точки /checkupdate	8
7.1.2 Відповідь кінцевої точки /checkupdate	8
7.1.3 Приклад Postman для кінцевої точки /checkupdate	9
7.2 Кінцева точка хмарного API	10
7.2.1 Запит кінцевої точки /putmeasure	10
7.2.2 Відповідь кінцевої точки /putmeasure	10
7.2.3 Приклад Postman для кінцевої точки /putmeasure	11
7.2.4 Запит кінцевої точки /getsources	12
7.2.5 Відповідь кінцевої точки /getsources	12
7.2.6 /getsources Endpoint Postman Приклад	13

1 Вступ

Цей документ є довідковим матеріалом для API синхронізації PoolLab2 через Wi-Fi (WSAPI). Він призначений для бажаючих інтегрувати PoolLab2 у свій власний веб-сервер хмари, щоб автоматично отримувати дані вимірювань від PoolLab2 без використання хмарного сервісу LabCom.



2 WiFi Інтерфейс

PoolLab2 оснащений вбудованим радіоприймачем з підтримкою WiFi 4, який можна використовувати для оновлення прошивки та бази даних пристрою, а також для синхронізації вихідних та вимірювальних даних із веб-сервером. Перш ніж використовувати будь-які функції WiFi, необхідно виконати одноразове налаштування через Bluetooth LE API. Як мінімум, необхідно встановити SSID та пароль точки доступу WiFi.

2.1 Взаємодія з користувачем

За промовчанням радіозв'язок вимкнено. WiFi-інтерфейс повинен бути увімкнений кінцевим користувачем на пристрої. Клавіша на клавіатурі-кришці вказує, яку клавішу слід натиснути для запуску **WiFi-процесу**. Після завершення процесу WiFi-радіо вимикається автоматично.

3 Вимоги до WiFi та мережі

3.1 Version and Security

Для роботи PoolLab2 потрібна точка доступу WiFi 4, налаштована на безпеку WPA2 та пароль з макс. 32 символи. Незахищені мережі та порожні паролі не підтримуються.

3.2 (B)SSID

Точка доступу не повинна надсилати свій SSID . Однак SSID не повинен бути порожнім та містити лише символи ASCII.

3.3 IP / DHCP

Точка доступу повинна автоматично призначати IPv4-адресу PoolLab2 за протоколом DHCP. Статичні IP-адреси не підтримуються.

3.4 NTP

На етапі **WiFi-процесу** вбудоване ПЗ періодично намагається синхронізувати свій внутрішній годинник з сервером часу. Точка доступу може надати PoolLab2 мережевий час через DHCP-параметри 004 та 042. Якщо вона не надана, PoolLab2 спробує підключитися до сайту time.nist.gov.

4 Вимоги до серверу / криптографії

Веб-сервер, здатний обслуговувати дійсні кінцеві точки **Cloud API** або **Update API**, повинен забезпечувати безпечне з'єднання через веб-сокети.

4.1 Socket

Сервер повинен забезпечити з'єднання TCP/IPv4 на порту 443.

4.2 SSL

Вбудоване програмне забезпечення вимагає, щоб кінцева точка API забезпечувала безпеку TLS/SSL. Сервер повинен забезпечувати шифрування на основі дійсного SSL-сертифіката, який відповідає URL-адресі, за якою він запитується. Сертифікат має бути підписаний довіреним кореневим сертифікатом. В даний час прошивка надає набір довірених корневих сертифікатів, витягнутих із сховища сертифікатів Mozilla CA, а також кореневий сертифікат Let's Encrypt.

4.3 WebSocket

Корисні дані в протокол WebSocket/HTTP, тобто. GET/POST-запити. **Кінцева точка Update API** має підтримувати заголовок запиту Range HTTP.

5 Конфігурація

Перш ніж використовувати WiFi-інтерфейс PoolLab2, необхідно виконати деякі налаштування за допомогою API Bluetooth LE. Докладніше про API Bluetooth розказано в окремому документі PoolLab2 BLE API Reference. У цьому розділі наведено лише огляд того, що можна налаштувати, про те, як встановити конфігурацію, читайте у вищезгаданому документі.

5.1 WiFi Логіни

Пара з SSID та пароля точки доступу WiFi. Див. PoolLab2 BLE API Reference [5.17 SET WIFI LOGINS Command].

5.2 Оновлення шляху API

URL-адреса, на якій розміщено кінцеву точку самооновлення. Див. PoolLab2 BLE API Reference [5.9 SET UPDATE API PATH Command]. За замовчуванням вказує канал стабільного випуску оригінального виробника пристрою [Water-i.d. GmbH].

5.3 Шлях до хмарного API

URL-адреса, на якій розміщено кінцеву точку синхронізації даних ("Cloud"). Див. PoolLab2 BLE API Reference [5.10 SET CLOUD API PATH Command]. За промовчанням вказується "Cloud LabCom" (<https://cloud-synchronizer.labcom.cloud/api/v1>).

5.4 Логіни хмарного API

Ім'я користувача та пароль для входу в налаштовану кінцеву точку Cloud API. Див. PoolLab2 BLE API Reference [5.18 SET CLOUD LOGIN Command]. Для "Cloud LabCom" ви можете безкоштовно зареєструвати обліковий запис на сайті <https://labcom.cloud/> та використовувати ці логіни для PoolLab2.

6 WiFi-процес

Коли кінцевий користувач запускає **WiFi-процес**, прошивка пристрою виконує такі дії:

Етап	Коментар
Battery Check	Перериває роботу, якщо рівень заряду батареї занадто низький для надійної активації радіоінтерфейсу
Radio Power On	Активує вбудоване радіо
WiFi AP Scan	Сканування на наявність налаштованої точки доступу
WiFi Connect	Підключіться до налаштованої точки доступу
DHCP Handshake	Запит та узгодження IPv4-адреси зі шлюзом
NTP Update	Синхронізація внутрішнього годинника з NTP-сервером шлюзу
Cloud Synchronization	Тільки якщо налаштовані дійсні дані для входу в систему
Update Check	Завантажує інформацію про доступну прошивку та версію бази даних
Update Dialog	Екранний діалог підтвердження для кінцевого користувача
Database Update	Самостійно оновлюйте базу даних з хімії
Firmware Update	Самостійне оновлення прошивки

Таблиця 1: Етапи WiFi-процесу

7 Кінцеві точки API

Під час **WiFi-процесу** вбудоване програмне забезпечення використовує різні кінцеві точки API. Кінцевий користувач може налаштувати URL кінцевих точок, наприклад, для синхронізації даних вимірювань з системою даних користувача. Щоб отримати додаткові відомості про те, як змінити адреси кінцевих точок API, див. *Configuration/Конфігурація*.

7.1 Кінцева точка API оновлення

Змінюючи адресу кінцевої точки/(Endpoint address) API Update, клієнт може перенаправити перевірку оновлень та самооновлення прошивки на власний сервер. Однак прошивка користувача/модифікована не може бути встановлена на пристрій, оскільки завантажувач відкидає непідписані двійкові файли. Мета дозволу кінцевої точки користувача API Update полягає в тому, щоб дозволити клієнтам запускати локально кешовані копії оригінальної кінцевої точки/(Endpoint address) виробника.

Заводське значення за замовчуванням [`cloud-synchronizer.labcom.cloud/api/v1`]. Префікс `https://` автоматично додається вбудованим програмним забезпеченням і не повинен включатися при переході на нову URL-адресу. Це **базова URL** для кінцевих точок API оновлення (`Update API Endpoints`).

7.1.1 Запит кінцевої точки `/checkupdate`

URL кінцевої точки : [`base-url`]/`checkupdate`

Тип запити: POST

Параметри: `oem id`, `hwrev`, `fw version` (`x-www-form-urlencoded`)

`oem id` буде встановлено ціле значення, що вказує на OEM-версію прошивки. Зверніть увагу, що для кожної OEM-версії існує файл прошивки, тому серверу необхідно обробити цю змінну, щоб запропонувати у відповідь правильний URL завантаження прошивки.

`hwrev` буде встановлено ціле значення, що вказує на апаратну ревізію пристрою, що виконує запит. І знову сервер повинен обробити цю змінну, щоб надати відповідну URL-адресу завантаження файлу образу прошивки.

`fw version` буде встановлено поточне значення версії мікропрограми пристрою, що виконує запит. Сервер повинен обробити цю змінну, щоб обмежити версію бази даних, запропоновану як оновлення пристрою. Для отримання додаткової інформації надішліть запит на адресу `leon.hock@water-id.com`.

7.1.2 Відповідь кінцевої точки `/checkupdate`

При виникненні помилки на стороні сервера може бути повернено будь-який http-код помилки для переривання процесу на стороні пристрою. Якщо запит неправильно сформований (відсутня будь-який параметр, параметр неправильного типу тощо), сервер повинен відповісти http-кодом 200 OK та вмістом тіла, встановленим у [`S:ERR`] (без дужок). Це призведе до переривання процесу оновлення, а прошивка покаже код помилки "update protocol error".

У разі успіху сервер відповідає http-кодом 200 OK та вмістом тіла, що має вигляд [`S:OK;A;B;X;Y`] (без дужок).

Де **A** та **B** - цілі числа, що вказують (**A**) код версії доступного оновлення прошивки, а також (**B**) код версії доступного оновлення бази даних.

X має бути повністю визначеною URL-адресою (**X**) файлу оновлення мікропрограми, що відповідає запитаній `hardware revision` та OEM ID, а також коду версії, переданому разом із відповіддю.

Y має бути повністю визначеною URL-адресою сховища оновлень бази даних, що відповідає запитаній `hardware revision` та OEM ID, а також коду версії бази даних, зазначеному у відповіді. Ця URL-адреса використовується як `database-base-url`. Мікропрограма додає до цього шляху кілька токенів, якими вона очікує отримати http-файли для завантаження файлів оновлень.

Шлях до файлу / Ім'я файлу	Коментар
[database-base-url]/pdb_parameter.txt	Parameter file
[database-base-url]/tlist_0_tab.txt	T-Menu 0
[database-base-url]/tlist_1_tab.txt	T-Menu 1
[database-base-url]/tlist_2_tab.txt	T-Menu 2
[database-base-url]/tlist_0_liq.txt	T-Menu (liq) 0
[database-base-url]/tlist_1_liq.txt	T-Menu (liq) 1
[database-base-url]/tlist_2_liq.txt	T-Menu (liq) 2
[database-base-url]/tlist_1_glass.txt	T-Menu (glass) 1
[database-base-url]/pdb_units.txt	Units file
[database-base-url]/pdb_curves.txt	Curves file

Таблиця 2: Файли поновлення бази даних

Структура та вміст цих файлів публічно не документовані. Розробникам, які бажають надати копії сервера оновлень, що локально кешуються, рекомендується надати копію структури каталогів і файлів, що знаходяться на офіційному сервері оновлень (використовуючи заводську URL кінцевої точки за замовчуванням).

7.1.3 Приклад Postman для кінцевої точки /checkupdate

The screenshot shows a Postman request configuration for a POST method to the URL `https://cloud-synchronizer.labcom.cloud/api/v1/checkupdate`. The request body is set to `x-www-form-urlencoded` and contains the following data:

Key	Value	Bulk Edit
<input checked="" type="checkbox"/> oem_id	1	
<input checked="" type="checkbox"/> hwrev	1	
<input checked="" type="checkbox"/> fw_version	4	
Key	Value	

The response is a 200 OK status with a response time of 108 ms and a body size of 386 B. The response body is displayed in HTML format:

```
1 S:OK;5;10;https://lab.studio/updater/poollab/release/1/firmware.bin;https://lab.studio/updater/poollab/release/1/pdb
```

7.2 Кінцева точка хмарного API

PoolLab2 може синхронізувати дані вимірювань та джерела з веб-сервером одним натисканням кнопки. За промовчанням таким веб-сервером є безкоштовний LabCom Cloud (<https://labcom.cloud>). Змінивши URL Cloud API Endpoint, кінцевий користувач може налаштувати PoolLab2 на синхронізацію даних вимірювань з користувацьким веб-сервером / хмарним сервісом.

Заводське значення за замовчуванням [cloud-synchronizer.labcom.cloud/api/v1]. Префікс <https://> автоматично додається до вбудованого програмного забезпечення і не повинен вмикатися при переході на нову URL-адресу. Це **базова** URL для Cloud API Endpoints.

7.2.1 Запит кінцевої точки /putmeasure

Ця кінцева точка Cloud API Endpoint використовується вбудованим програмним забезпеченням для завантаження збережених даних *Measurement Data* (вимірювань з пристрою). Кожен запис вимірювань завантажується в рамках окремого запиту, тому рекомендується використовувати сервер, що підтримує постійне з'єднання HTTP, також відоме як HTTP keep-alive, для об'єднання декількох запитів в рамках одного сокетного з'єднання.

URL кінцевої точки: [base-url]/putmeasure

Тип запиту: POST

Параметри: email, pass, serial, scenario id, parameter id, value, time_taken, account_id, account_uuid (x-www-form-urlencoded)

Параметри **email** та **pass** будуть встановлені відповідно до логіну хмарного облікового запису, налаштованого на пристрої. Користувальницький сервер може ігнорувати ці значення, проте прошивка вимагає, щоб внутрішня конфігурація мала непусті значення для електронної пошти та паролів входу. В іншому випадку прошивка вважає, що логіни не налаштовані і не намагається синхронізувати хмарні дані.

Параметр **serial** буде встановлений у текстове представлення серійного номера пристрою, який запитує.

Параметри **scenario id** та **parameter id** вказують, який тип виміру було зроблено. За додатковою інформацією звертайтеся за адресою leon.hock@water-id.com.

Параметр **value** представлятиме значення вимірювання у форматі з плаваючою точкою. Значення 1000000.0 означає переповнення діапазону або так звану помилку "Overrange". Значення -1000000.0 означає переповнення діапазону або так звану помилку "Underrange".

Параметр **time taken** - це тимчасова мітка часу (UTC), коли було здійснено вимір.

Параметри **account id** та **account uuid** встановлюються у значення *Source ID* та *Source UUID* джерела, яке було вибрано при виконанні даного вимірювання.

7.2.2 Відповідь кінцевої точки /putmeasure

При виникненні помилки на стороні сервера може бути повернено будь-який http-код помилки для переривання процесу на стороні пристрою. Якщо запит неправильно сформований (відсутня будь-який параметр, параметр неправильного типу тощо), сервер повинен відповісти http-кодом 200 OK та вмістом тіла, встановленим у [S:ERR] (без дужок). Це призведе до переривання процесу синхронізації, а прошивка покаже код помилки "cloud protocol error".

У разі успіху сервер повинен відповісти http-кодом 200 OK та вмістом блоку, встановленим у [S:OK] (без дужок).

7.2.3 Приклад Postman для кінцевої точки /putmeasure

The screenshot shows the Postman interface for a POST request. The URL is `https://cloud-synchronizer.labcom.cloud/api/v1/putmeasure`. The request body is set to `x-www-form-urlencoded` and contains the following data:

Key	Value
email	hock@pool-id.com
pass	d054ce5
serial	P2123457678
scenario_id	1
parameter_id	1
value	0.42
time_taken	1704218856
account_id	133
account_uuid	00000000000000000000000000000000

The response is a 200 OK status with a response time of 118 ms and a body size of 342 B. The response body is currently empty, showing only the status `1 S:OK`.

7.2.4 Запит кінцевої точки /getsources

Ця кінцева точка Cloud API використовується вбудованим програмним забезпеченням для синхронізації списку джерел, збережених на пристрої. Джерело - це один з елементів, що вибираються користувачем на пристрої, який використовується для організації даних вимірювань за місцезнаходженням або клієнтом. Мікропрограма завжди синхронізує дані вимірювань у першу чергу, а Джерела - в другу, щоб не було даних вимірювань з незв'язаними ідентифікаторами Джерел у разі, якщо одне або кілька Джерел будуть видалені між синхронізаціями.

Джерело складається з числового (32-бітного) ідентифікатора джерела, 32-байтного UUID джерела та 23-символьного (ASCII) імені джерела. На пристрої відображається лише ім'я джерела.

URL кінцевої точки: [base-url]/getsources

Тип запиту: POST

Параметри: email, pass, serial (x-www-form-urlencoded)

Параметри **email** та **pass** будуть встановлені відповідно до логіну хмарного облікового запису, налаштованого на пристрої. Користувальницький сервер може ігнорувати ці значення, проте прошивка вимагає, щоб внутрішня конфігурація мала непусті значення для електронної пошти та паролів входу. В іншому випадку прошивка вважає, що логіни не налаштовані і не намагається синхронізувати хмарні дані.

Параметр **serial** буде встановлений у текстове представлення серійного номера пристрою, який запитує. LabCom Cloud використовує серійний номер для того, щоб можна було налаштувати різні набори джерел для різних пристроїв, використовуючи при цьому ті самі дані для входу в хмарний обліковий запис. Користувальницький сервер може ігнорувати це значення.

7.2.5 Відповідь кінцевої точки /getsources

При виникненні помилки на стороні сервера може бути повернено будь-який http-код помилки для переривання процесу на стороні пристрою. Якщо запит неправильно сформований (відсутня будь-який параметр, параметр неправильного типу тощо), сервер повинен відповісти http-кодом 200 OK та вмістом тіла, встановленим у [S:ERR] (без дужок). Це призведе до переривання процесу синхронізації, і прошивка покаже код помилки "помилка хмарного протоколу".

У разі успіху сервер повинен відповісти http-кодом 200 OK та вмістом тіла, встановленим на список Джерел, які мають бути збережені на пристрої. Відповідь повинна починатися з [S:OK], за яким слідує символ нового рядка (\n). Після цього відповідь має бути списком усіх Джерел. Кожне Джерело складається з 3 значень, розділених точкою з комою (;) (Ім'я Джерела; UUID Джерела; ID Джерела), за якими слідує символ нового рядка (\n). Останнє Джерело у відповіді не повинно супроводжуватись символом нового рядка.

Повинне бути вказане хоча б одне джерело, інакше мікропрограма очистить усі поточні джерела та автоматично додасть джерело за промовчанням.

7.2.6 /getsources Endpoint Postman Приклад

POST [Send](#)

Params Auth Headers (8) **Body** Pre-req. Tests Settings [Cookies](#)

x-www-form-urlencoded

	Key	Value	Bulk Edit
<input checked="" type="checkbox"/>	email	hock@pool-id.com	
<input checked="" type="checkbox"/>	pass	d054cr7...39f...351d0f...Lzob32ae9...	
<input checked="" type="checkbox"/>	serial	P2123457678	
	Key	Value	

Body [200 OK](#) 390 ms 374 B [Save Response](#)

Pretty Raw Preview Visualize **HTML**

```
1 S:OK
2 Default Sampling Point;000000000000000000000000000000;133
3 Default Sampling Point;000000000000000000000000000000;134
4 Default Sampling Point;000000000000000000000000000000;135
5 Default Sampling Point;000000000000000000000000000000;136
6 Test Account;000000000000000000000000000000;139
```