Obsah

Přihlášení jednotky po startu (POST /login) **1**
K čemu slouží 1
Spouštěč a opakování 1
Cíl a gating 1
Formát požadavku 1
Příklad těla 2
Odpověď serveru 2
Stav na stránce Info 3
Zdrojové soubory 3

Přihlášení jednotky po startu (`POST /login`)

Platí pro verzi NPA R04: 2.15.beta9 · Vytvořeno: 2026-09-04

Po každém startu softwaru jednotka pošle na cloudový/centrální server zprávu **login**: identifikaci jednotky, čas, verzi firmware a **celý konfigurační JSON**. Zprávu opakuje, dokud ji server nepotvrdí. Funkce je obdobou odesílání měřených hodnot (`measuring`, viz `interfaces.md` §7) a konfigurace (`config-upload.md`), liší se spouštěčem, endpointem a tím, že **vyhodnocuje odpověď serveru**.

> **Dostupné od verze `2.15.beta9`.**

K čemu slouží

- Server se dozví o každém startu / restartu jednotky (včetně důvodu restartu v `uptime`).
- Server má po startu vždy aktuální config i verzi firmware, i když se od posledního uložení nic nezměnilo (`/config` chodí jen při uložení a jen při změně obsahu).
- Potvrzení serverem („ok“) dává jednotce jistotu, že zpráva dorazila; do té doby ji opakuje.

Spouštěč a opakování

- Stavový automat v `NRemoteLog::Loop()` (tick 500 ms): hned po startu (první tick) se pošle `SendLogin()`. Když server **nepotvrdí** (chyba sítě, HTTP chyba, odpověď bez `"res":"ok"`, nebo VPN ještě neběží), pošle se znovu **za 60 s**; bez omezení počtu pokusů.
- Po potvrzení se za běhu procesu už neposílá. Další login vznikne až po restartu SW.
- Login běží **nezávisle** na stahování posledního stavu (`/getlastmeas`) a odesílání měření (`/measuring`) – ty nečekají na potvrzení loginu. Starší server bez endpointu `/login` tedy nic nezablokuje (jednotka jen každou minutu opakuje pokus).

Cíl a gating

- **Cíl:** IP serveru z VPN rozhraní (`GetVpnNpaServerIp()`, `interfaces.md` §9). Když je prázdná (tunel neběží), pokus se přeskočí a zopakuje za minutu.
- **Gating:** jen pokud `mC2["ns_cloud"] != "false"` (stejné jako `measuring` a `config`).
- **Endpoint:** `POST http://<vpn-srv>/login`, `Content-Type: application/json`.

Formát požadavku

Klíč	Typ	Význam
`sn`	string	sériové číslo jednotky (`NConfig::mSN`)
`name`	string	název výrobny (`mC2["bs_name"]`, Základní nastavení)
`time`	string	čas odeslání (`GetUTCTime()`, ISO 8601)

Klíč	Typ	Význam
`plc_uuid`	string	UUID jednotky (`NConfig::plc_uuid`)
`build`	string	verze firmware (`G_BUILD`, např. `2.15.beta9`)
`build_date`	string	datum a čas sestavení (`G_BUILD_DATE`)
`build_num`	number	číselná verze pro porovnání (`G_BUILD_NUM`)
`uptime`	string	doba běhu + důvod posledního restartu, jako na stránce Info (`NUptime::GetUptime()`)
`ip`	string	IP adresa jednotky v LAN
`ip_vpn`	string	IP adresa jednotky ve VPN
`attempt`	number	pořadí pokusu od startu (1, 2, …)
`config`	object	**celý** konfigurační JSON – přesně to, co se ukládá na disk a posílá v `/config` (`NConfig::ConfigJson()`)
`config_ok`	bool	výsledek kontroly konfigurace (`NConfig::ConfigWarnings()` prázdné)
`warnings`	array	seznam varování konfigurace (texty jako na stránce Info)

Objekt `config` má stejnou strukturu jako v `config-upload.md` (`mSN`, `mUsers`, `mAdmins`, `mGurus`, `mC2`, `plc_uuid`, `mTimeLogSec`, `last_save`).

> **Pozn.:** stejně jako `/config` obsahuje payload i citlivé části konfigurace (uživatelské účty). Přenos běží přes VPN tunel.

Příklad těla


```
{
  "sn": "R04-0001",
  "name": "FVE Horní Lhota",
  "time": "2026-09-04T12:56:16+02:00",
  "plc_uuid": "{....}",
  "build": "2.15.beta9",
  "build_date": "Sep  4 2026 12:40:11",
  "build_num": 215009,
  "uptime": "00:00:01 od watchdog",
  "ip": "192.168.1.20",
  "ip_vpn": "10.80.3.12",
  "attempt": 1,
  "config": { "mSN": "R04-0001", "mC2": { "device": [ ], "location": [ ] }, "plc_uuid": "{...
  "config_ok": true,
  "warnings": []
}
```


Odpověď serveru

Server potvrdí přijetí JSON odpovědí:


```
{ "res": "ok" }
```


Jen tato odpověď (HTTP úspěch **a** `res == "ok"`, stejná konvence jako odpovědi JSON-RPC) ukončí opakování. Cokoli jiného (HTTP 200 s jiným tělem, chybový kód, timeout, nedostupný server) se bere jako nepotvrzeno a za 60 s jde další pokus s vyšším `attempt`. Do logu jednotky se píše `login: pokus N`, `login ok (pokus N)`, případně důvod neúspěchu.

Stav na stránce Info

Řádek **Login na server (po startu)** na stránce *Info* (a řádek `Login na server` v diagnostice *Kopírovat diagnostiku*) ukazuje:

- zeleně `ok (potvrzeno serverem <čas>, pokus N)` – server potvrdil;
- `ceka na potvrzeni serverem, odeslano Nx, naposledy <čas>; dalsi pokus za S s` – zatím nepotvrzeno; po neúspěchu červeně s doplňkem `CHYBA: …` (např. `server nepotvrdil (HTTP 200, res != ok): {}`, `chyba odeslani: Connection refused`, `VPN server IP neznama (tunel tun_npa_r04 nebezi)`);
- `vypnuto (ns_cloud = false)` – odesílání do cloudu je vypnuté.

Zdrojové soubory

Část	Soubor / symbol
Stavový automat a opakování	`nremotelog.cpp` → `NRemoteLog::Loop()` (`mLoginState`, `NRL_LOGIN_RETRY_TICKS`)
Sestavení a odeslání zprávy	`nremotelog.cpp` → `NRemoteLog::SendLogin()`
Config jako JSON (společné pro disk, `/config`, `/login`)	`nconfig.cpp` → `NConfig::ConfigJson()`
Globální instance	`main.cpp` → `NRemoteLog gNRemoteLog`