Obsah

Odesílání konfigurace na server (POST /config) **1**
Princip . . . 1
Spouštěč . . . 1
Cíl a gating . . . 1
Formát požadavku . . . 1
Příklad těla . . . 2
Odpověď serveru . . . 2
Bezpečné ukládání configu na disk . . . 3
Zdrojové soubory . . . 3

Odesílání konfigurace na server (POST /config)

Platí pro verzi NPA R04: 2.13.beta20 · Vytvořeno: 2026-06-17

Tento dokument popisuje, jak jednotka NPA R04 při každém uložení konfigurace odešle celý konfigurační JSON na cloudový/centrální server. Funkce je obdobou odesílání měřených hodnot (`measuring`, viz `interfaces.md` §7), ale s jiným spouštěčem a endpointem.

> **Dostupné od verze `2.13.beta20`.** Spolu s bezpečným ukládáním configu na disk přidáno v této verzi (viz `changelog.txt`).

Princip

Při **každém uložení konfigurace** (`NConfig::Save()`) proběhnou dva kroky v tomto pořadí:

1. **Bezpečný zápis na disk** – konfigurace se zapíše do nového číslovaného souboru `/data/NPA_R04.cnf.<N>` + sidecar `.sum` s kontrolní sumou (viz Bezpečné ukládání configu).
2. **Odeslání na server** – až po úspěšném zápisu na disk se celý konfigurační JSON odešle na server. Odeslání je **asynchronní** (běží na pozadí přes `QNetworkAccessManager::post`), takže `Save()` neblokuje.

Pořadí je záměrné: nejdřív musí být konfigurace bezpečně na disku, teprve potom se posílá. Když odeslání selže (server nedostupný, chyba sítě), na lokálně uloženou konfiguraci to nemá vliv.

Spouštěč

Odeslání se vyvolá z `NConfig::Save()` voláním `gNRemoteLog.SendConfig(mSet)`, kde `mSet` je přesně ten JSON objekt, který se zapisuje na disk. `Save()` se volá při jakékoli změně konfigurace – z webového rozhraní (přidání/úprava zařízení, lokace, nastavení sítě, factory settings…), z JSON-RPC i z interní logiky. Posílá se tedy při **každé** změně konfigurace, ne periodicky.

Cíl a gating

- **Cíl:** IP serveru se zjistí z VPN rozhraní přes `GetVpnNpaServerIp()` (viz `interfaces.md` §9). Když je prázdná (VPN/server nedostupný), odeslání se přeskočí.
- **Gating:** posílá se pouze pokud `mC2["ns_cloud"] != "false"` (stejné gate jako u `measuring`).
- **Endpoint:** `POST http://<vpn-srv>/config`, `Content-Type: application/json`.

Formát požadavku

Tělo POSTu je JSON objekt:

Klíč	Typ	Význam
`sn`	string	sériové číslo jednotky (`NConfig::mSN`)
`time`	string	čas odeslání v UTC (`GetUTCTime()`)
`plc_uuid`	string	UUID jednotky (`NConfig::plc_uuid`)
`config`	object	**celý** konfigurační JSON (obsah ukládaného souboru, `NConfig::ConfigJson()`)
`config_ok`	bool	výsledek kontroly konfigurace (`NConfig::ConfigWarnings()` prázdné)
`warnings`	array	seznam varování konfigurace (texty jako na stránce Info)

Objekt `config` odpovídá obsahu `/data/NPA_R04.cnf.<N>` a obsahuje mj.:

- `mSN` – sériové číslo
- `mUsers`, `mAdmins`, `mGurus` – uživatelské účty
- `mC2` – hlavní konfigurace (zařízení, lokace, modbus mapování, výkonové úrovně…)
- `plc_uuid`, `mTimeLogSec`, `last_save`

Pozn.: payload obsahuje i citlivé části konfigurace (uživatelé/hesla v `mUsers`/ `mAdmins`/`mGurus`). Přenos běží přes VPN tunel.

Příklad těla

```
{
  "sn": "R04-0001",
  "time": "2026-06-17T14:11:00",
  "plc_uuid": "{....}",
  "config": {
    "mSN": "R04-0001",
    "mUsers": { },
    "mAdmins": { },
    "mGurus": { },
    "mC2": { "device": [ ], "location": [ ] },
    "plc_uuid": "{....}",
    "mTimeLogSec": 60,
    "last_save": "2026-06-17T14:11:00"
  }
}
```

Odpověď serveru

Odpověď se aktuálně nevyhodnocuje – po dokončení požadavku se reply jen uklidí (`deleteLater()`), případná chyba se zaloguje (`qDebug() << "SendConfig error:" …`). Odeslání je „fire-and-forget".

Stejný objekt `config` (a `config_ok`/`warnings`) se posílá i po každém startu SW ve zprávě **login** (`POST /login`), kterou jednotka opakuje, dokud ji server nepotvrdí – viz `login.md`.

Bezpečné ukládání configu na disk

Kontext kroku 1 (zápis na disk), na kterém odeslání závisí:

- Místo přepisu jednoho souboru se ukládá nový soubor `/data/NPA_R04.cnf.<N>`, kde `N` je o `1` vyšší než index naposledy načteného configu.
- Ke každému souboru se zapisuje sidecar `/data/NPA_R04.cnf.<N>.sum` se SHA-256 kontrolní sumou (zapisuje se až po datech – nedopsaný soubor bez `.sum` se při načtení přeskočí).
- Na disku se drží posledních **10** souborů; při překročení se maže ten s nejmenším číslem.
- Při načítání se bere soubor s nejvyšším číslem a ověří se suma; při nesedící/chybějící sumě se zkouší nižší číslo.
- Legacy `/data/NPA_R04.cnf` (bez čísla a sumy) se bere jako **index 0** → zpětná kompatibilita se starším systémem.

Zdrojové soubory

Část	Soubor / symbol
Spuštění odeslání	`nconfig.cpp` → `NConfig::Save()`
Bezpečný zápis na disk	`nconfig.cpp` → `WriteConfigData()`, `ReadConfigData()`, `PruneConfigs()`
Odeslání na server	`nremotelog.cpp` → `NRemoteLog::SendConfig()`
Globální instance	`main.cpp` → `NRemoteLog gNRemoteLog`