

Obsah

Port 8080 — HTTP/HTML webserver	1
Parametry spojení	1
Zpracování požadavku (NHttpSocket::MsgReady)	1
Relace a autentizace (NSession)	1
GET endpointy	2
Stránky (HTML)	2
JSON data (AJAX)	2
Statické soubory	2
Konfigurace lokací a zařízení	2
Ruční ovládání a testy	2
Povely regulace (ruční, z webu)	3
Systém	3
POST endpointy	3
Šablony	3
Pozn. k nasazení	4

Port 8080 — HTTP/HTML webserver

Platí pro verzi NPA R04: 2.13.beta20 · Vytvořeno: 2026-06-02

Zdroj: server.cpp / server.h (třída Server, singleton Server::GetServer()), nhttpsocket.{h,cpp} (parsování požadavku), nhtml.{h,cpp} (skládání odpovědí), nsession.{h,cpp} (relace/oprávnění).

Konfigurační a monitorovací webové rozhraní jednotky. Surový QTcpServer (ne QHttpServer) s ručním parsováním HTTP.

Parametry spojení

Parametr	Hodnota	Zdroj
Naslouchá na	0.0.0.0	pevně
TCP port	8080	pevně (Server::Init)
Socket	NHttpSocket (potomek QTcpSocket)	

Příchozí spojení (incomingConnection) připojí slot readClient() na readyRead.

Zpracování požadavku (NHttpSocket::MsgReady)

- Požadavek se akumuluje do mMsg, zpracuje se po ≥ 3 řádcích.
- mUrl = první řádek (percent-decoded), tj. GET /a/b/<command>?<param> HTTP/1.1.
- Příkaz** = 4. segment cesty: commands = path.split("/"), command = commands[3]. Parametry za ? jsou v pagess[1], vícenásobné odděleny &.
- mSession = hodnota cookie NsessionId (jinak se vygeneruje nové UUID).
- mEAUser = hodnota ea_rolé z cookie (externí autentizace).
- mPost = tělo požadavku (dle Content-Length, percent-decoded, + → mezera).

Relace a autentizace (NSession)

- Stav drží NSession (mUserName, mAuthLevel, mEditLocationUuid, mEditDeviceUuid, mIoLocationUuid ...); web jej vyzvedne přes mNHtml.SetSession(mSession, mEAUser).
- Cookie NsessionId se nastavuje v každé HTML hlavičce (HtmlHead): Set-Cookie: NsessionId=<id>; Path=/.

- Přihlášení: POST login.html → NSession::Login(post); odhlášení: logout.html; změna hesla: POST password.
- Typy odpovědí: HtmlHead (text/html), JsonHead (application/json), JsHead, CssHead.

GET endpointy

Stránky (HTML)

index.html, index2.html, info.html, net_settings.html, basic_settings.html, factory_settings.html, settings.html, regulation.html, locations_dis.html, devices_detection.html, io_settings.html, param.html, config.html, data.html, state.html, log.html, debug.html, public.html, modbus_registers.html, reboot.html, restart.html, login.html, logout.html

JSON data (AJAX)

Endpoint	Obsah
get_index_data	hlavní přehled (NState::GetInxeDataJson)
get_bin?<uuid>	binární I/O lokace (GetInOutJson)
get_da	stav D/A výstupů (GetDaJson)
getstate	stav (pole p_inv/q_inv)
getid	sériové číslo + příznak -unregistered
update_gaude	data pro měřidla (gauge)
info.html (JSON větev)	info + test HTML bloky

Statické soubory

style.css, npa.css, fonts.css, main.js, main1.js, gauge.min.js (servírováno z adresáře html/).

Konfigurace lokací a zařízení

Endpoint	Akce
lo_add / lo_edit?<uuid> / lo_delete?<uuid>	přidat / editovat / smazat lokaci
lo_delete_device?<uuid>	odebrat zařízení z lokace
dd_add / dd_add_tcp / dd_add_485	přidat zařízení (manuálně / TCP / RS-485)
dd_edit?<uuid> / dd_delete	editace / smazání zařízení
setting_next	přidat prázdné zařízení do seznamu
io_select_location	výběr lokace pro I/O
re_add_accu	přidat akumulátor

Ruční ovládání a testy

Endpoint	Akce
releclick?<true/false>	sepnout/rozepnout relé (NIOPins::SetRele)
tranzistorclick?<true/false>	tranzistorový výstup (NIOPins::SetTranzistor)
rs485click?<true/false>	připojit RS-485 linku (NRs485::SetConnect)
setda?<n>&<value>	nastavit D/A výstup n (test, 30 cyklů)
settestbatt?<%>	testovací SOC baterie
outtoggle?<key>&<bit>	přepnout binární výstup
dd_test_out?<uuid> / dd_xch_out	test výstupů zařízení
instr?<uuid>&<fn>&<var>&<addr>&<param>	vložit Modbus instrukci do fronty driveru

Endpoint	Akce
clear_fram	smazat perzistentní stav (mSavedM)

Povely regulace (ruční, z webu)

Endpoint	Funkce NState	Úroveň
setglobalpower?<W>	SetGlobalRegPower	globální
setgloballevelp?<n>	SetGlobalRegLevel (stupeň P)	globální
setgloballevelq?<n>	SetGlobalQRegLevel	globální
setgloballevelfi?<n>	SetGlobalFiRegLevel (cosφ)	globální
setglobalbesspower?<W>	SetGlobalRegPowerBess	globální BESS
setglobalbesslevelch?<n>	SetGlobalRegBessLevelCharge	globální BESS nabíjení
setglobalbessleveldisch?<n>	SetGlobalRegBessLevelDischarge	globální BESS vybíjení
setlocationpower?<W> / setlocationlevelp?<n>	SetLocationRegPower/Level	lokace
setlocbesspower / setlocbesslevelch / setlocbessleveldisch	SetLocUuidReg(...)	lokace BESS
setdevpower?<W> / setdevlevelp?<n>	SetDevRegPower/Level	zařízení
setdevbesspower / setdevbesslevelch / setdevbessleveldisch	(zakomentováno)	zařízení BESS

Globální/lokační povely volají NState::Save() (perzistence do state.cnf). Lokační/zařízenové povely míří na mEditLocationUuid / mEditDeviceUuid z aktuální relace.

Systém

lang?<id> (přepnutí jazyka), password, save, restart_save, ping, reboot.html, restart.html.

POST endpointy

Tělo (mPost) se předává příslušné funkci; konfigurační POST zapisují do NConfig a volají NConfig::Save() → /data/NPA_R04.cnf.

Endpoint	Akce
login.html	NSession::Login(post) → html/back.html
save	Save(post) (uložení formuláře konfigurace)
password	změna hesla (NHtml::BasicSettings(post))
restart_save	Save + NConfig::SaveNetConfig → restart
lo_save	NConfig::SaveLocation(uuid, ...)
lo_add_device	NConfig::LocationAddDeviceTo(uuid, ...)
dd_add / dd_add_tcp / dd_add_485	Save + Devices::AddDevice[/Tcp/485]
dd_save	Devices::SaveDevice(uuid, ...)

Šablony

Odpovědi skládá NHtml ze souborů v html/<lang>/ (cz/, en/, plus v2/). Jazyk dle NConfig::mLanguage.GetLang
Statika a sdílené šablony jsou přímo v html/.

Pozn. k nasazení

`systemctl install .sh` kopíruje `html/` vedle binárky do `/usr/local/bin/npa/`, takže relativní cesty `html/...` fungují z pracovního adresáře služby.