API Dokumentation

Die FNA-Steuerung stellt eine REST-API bereit, über die alle Funktionen der Ampelsteuerung programmatisch abgerufen und gesteuert werden können. Die API ist unter `http://<ip-adresse>/api/` erreichbar.

Übersicht der Endpunkte

Methode	Endpunkt	Beschreibung	PIN Auth	Device Auth
GET	`/api/status`	Aktuellen Status der Ampeln abrufen	X	
GET	`/api/cmd`	Verfügbare Befehle abrufen	X	X
POST	`/api/cmd/{cmd}`	Befehl ausführen	X	X
GET	`/api/config`	Alle Konfigurationen abrufen	X	
POST	`/api/config`	Neue Konfiguration erstellen	X	
GET	`/api/config/preset`	Aktive Voreinstellung abrufen	X	
GET	`/api/config/volume`	Lautstärke abrufen	X	
PUT	`/api/config/volume`	Lautstärke setzen	X	
GET	`/api/config/brightness`	Helligkeit abrufen	X	
PUT	`/api/config/brightness`	Helligkeit setzen	X	
PUT	`/api/config/nach`	Nachschießen-Anzahl setzen	X	
DELETE	`/api/config/nach`	Nachschießen deaktivieren	X	
GET	`/api/config/{id}`	Einzelne Konfiguration abrufen	X	
PUT	`/api/config/{id}`	Konfiguration aktualisieren	X	
DELETE	`/api/config/{id}`	Konfiguration löschen	X	

Methode	Endpunkt	Beschreibung	PIN Auth	Device Auth
GET	`/api/logs`	Protokolleinträge abrufen	X	
DELETE	`/api/logs`	Protokoll löschen	X	
GET	`/api/logs/training`	Trainings-Modus abrufen	X	
PUT	`/api/logs/training`	Trainings-Modus setzen	X	
GET	`/api/time`	Systemzeit abrufen	X	
POST	`/api/sync-time`	Systemzeit synchronisieren	X	
POST	`/api/auth/login`	Anmelden mit PIN		
GET	`/api/auth/status`	Status des Session (Auth)		
POST	`/api/auth/logout`	Abmelden		
PUT	`/api/auth/pin`	Login PIN ändern	X	
GET	`/api/devices/pending`	Liste der wartenden Geräte	X	
POST	`/api/devices/pending/{device_id}/approve`	Gerät zulassen	X	
DELETE	`/api/devices/pending/{device_id}`	Wartendes gerät löschen	X	
GET	`/api/devices/approved`	Liste der zugelassenen Geräte	X	
DELETE	`/api/devices/approved/{device_id}`	Zugelassenes Gerät löschen	X	

Status

`GET /api/status`

Gibt den aktuellen Betriebsstatus beider Ampeln zurück.

Antwort `200 OK`

```
{
  "a1": {
```

```
    "red": true,
    "yellow": false,
    "green": false,
    "time": 120,
    "group": "A"
  },
  "a2": {
    "red": false,
    "yellow": false,
    "green": true,
    "time": 120,
    "group": "B"
  },
  "run": true,
  "group_mode": false,
  "count_arrows": 3,
  "alt_mode": false,
  "shoot_break": false,
  "shoot_off_mode": false
}
```


Feld	Typ	Beschreibung
`a1` / `a2`	Objekt	Status von Ampel 1 und 2
`a1.red`	Boolean	Rotes Licht aktiv
`a1.yellow`	Boolean	Gelbes Licht aktiv
`a1.green`	Boolean	Grünes Licht aktiv
`a1.time`	Integer	Verbleibende Zeit in Sekunden
`a1.group`	String	Zugewiesene Gruppe (z. B. `"A"` oder `"B"`)
`run`	Boolean	Steuerung läuft aktuell
`group_mode`	Boolean	AB/CD-Gruppenmodus aktiv
`count_arrows`	Integer	Aktuell eingestellte Pfeilanzahl
`alt_mode`	Boolean	Alternierend-Modus aktiv
`shoot_break`	Boolean	Schießpause aktiv
`shoot_off_mode`	Boolean	Gibt an, ob Nachschießen oder Stechen eingestellt ist

Befehle

`GET /api/cmd`

Gibt zurück, welche Befehle aktuell verfügbar (ausführbar) sind.

Antwort `200 OK`

```
{
  "CommandStatus": {
    "START": true,
    "STOP": false,
    "SWITCH": false,
    "BREAK": false
  }
}
```

Befehl	Beschreibung
`START`	Schießsequenz starten
`STOP`	Schießsequenz stoppen
`SWITCH`	Gruppe wechseln (bei Alternierend-Modus)
`BREAK`	Schießpause einleiten

`POST /api/cmd/{cmd}`

Führt einen Steuerungsbefehl aus.

Pfadparameter

Parameter	Typ	Beschreibung
`cmd`	String	Befehl: `START`, `STOP`, `SWITCH` oder `BREAK`

Beispiel

```
POST /api/cmd/START
```

Antworten

Code	Beschreibung
`200 OK`	Befehl erfolgreich ausgeführt
`400 Bad Request`	Ungültiger oder nicht erlaubter Befehl
`500 Internal Server Error`	Serverfehler

Konfigurationen

Konfigurationen enthalten alle Einstellungen für einen Schießbetrieb (z. B. Zeit pro Pfeil, Pfeilanzahl, Modus). Es können mehrere Konfigurationen gespeichert werden.

`GET /api/config`

Gibt alle gespeicherten Konfigurationen zurück.

Antwort `200 OK` – Array von Konfigurationsobjekten (siehe Felder unten)

`POST /api/config`

Erstellt eine neue Konfiguration.

Anfrage-Body

```
{
  "name": "Halle 18m",
  "time_per_arrow": 40,
  "prepare_time": 10,
  "count_arrows": 3,
  "group_mode": false,
  "alt_single": false,
  "alt_team": false
}
```

Feld	Typ	Beschreibung
`name`	String	Bezeichnung der Konfiguration
`time_per_arrow`	Integer	Zeit pro Pfeil in Sekunden
`prepare_time`	Integer	Vorlaufzeit in Sekunden

Feld	Typ	Beschreibung
`count_arrows`	Integer	Anzahl der Pfeile pro Durchgang
`group_mode`	Boolean	AB/CD-Gruppenmodus aktivieren
`alt_single`	Boolean	Alternierend Einzeln aktivieren
`alt_team`	Boolean	Alternierend Mannschaft aktivieren

Antwort `200 OK` – Die erstellte Konfiguration inkl. `id` und `preset`-Flag

---

`GET /api/config/{id}`

Gibt eine einzelne Konfiguration anhand ihrer ID zurück.

`PUT /api/config/{id}`

Aktualisiert eine vorhandene Konfiguration. Body entspricht `POST /api/config`.

`DELETE /api/config/{id}`

Löscht eine Konfiguration anhand ihrer ID.

`GET /api/config/preset`

Gibt die aktuell als Voreinstellung markierte Konfiguration zurück.

Antworten

Code	Beschreibung
`200 OK`	Voreinstellung gefunden
`404 Not Found`	Keine Voreinstellung gesetzt

Lautstärke & Helligkeit

`GET /api/config/volume`

Gibt die aktuelle Lautstärke der Hupe zurück.

Antwort `200 OK`

```
{ "amp_horn_vol": 80 }
```

`PUT /api/config/volume?vol={wert}`

Setzt die Lautstärke der Hupe.

Abfrageparameter

Parameter	Typ	Pflicht	Beschreibung
`vol`	Integer	Ja	Lautstärkewert

`GET /api/config/brightness`

Gibt die aktuelle Helligkeit der Ampeln zurück.

Antwort `200 OK`

```
{ "amp_bright": 100 }
```

`PUT /api/config/brightness?bright={wert}`

Setzt die Helligkeit der Ampeln.

Parameter	Typ	Pflicht	Beschreibung
`bright`	Integer	Ja	Helligkeitswert

Nachschießen

`PUT /api/config/nach?count={anzahl}`

Aktiviert das Nachschießen mit der angegebenen Pfeilanzahl. Bei Alternierendem Modus wird hiermit das Stecken mit angegebener Schützenzahl aktiviert.

Parameter	Typ	Pflicht	Beschreibung
`count`	Integer	Ja	Anzahl der Pfeile für das Nachschießen / Schützen für Stechen

---

`DELETE /api/config/nach`

Deaktiviert das Nachschießen.

Protokoll (Logs)

`GET /api/logs`

Gibt alle Protokolleinträge zurück.

Antwort `200 OK`

```
[
  {
    "id": 1,
    "timestamp": "2026-04-03T10:00:00",
    "action": "START",
    "description": "Schießsequenz gestartet"
  }
]
```

Feld	Typ	Beschreibung
`id`	Integer	Eindeutige ID des Eintrags
`timestamp`	DateTime	Zeitstempel des Ereignisses
`action`	String	Ausgeführte Aktion
`description`	String	Beschreibung des Ereignisses

DELETE /api/logs

Löscht alle Protokolleinträge.

GET /api/logs/training

Gibt zurück, ob der Trainings-Modus aktiv ist (Protokolleinträge werden im Trainings-Modus nicht für das Schießleiterprotokoll gewertet).

Antwort `200 OK`

```
{ "training": false }
```

PUT /api/logs/training?training={wert}

Aktiviert oder deaktiviert den Trainings-Modus.

Parameter	Typ	Pflicht	Standard	Beschreibung
`training`	Boolean	Nein	`false`	`true` = Trainings-Modus aktiv

Zeit

GET /api/time

Gibt die aktuelle Systemzeit der Steuerung zurück.

Antwort `200 OK`

```
{ "time": "2026-04-03T10:00:00" }
```

POST /api/sync-time

Synchronisiert die Systemzeit der Steuerung.

Anfrage-Body

```
{ "time": "2026-04-03T10:00:00" }
```

Antwort `200 OK` – Bestätigung mit der gesetzten Zeit

Auth

`POST /api/auth/login`

Authentifiziert die UI

Anfrage-Body

```
{ "pin": "123456" }
```

Antwort `200 OK`

```
{  "token": "AA-BB-CC-DD" }
```

`GET /api/auth/status`

Gibt den Auth Status zurück

Antwort `200 OK`

```
{  "authenticated": true }
```

`POST /api/auth/logout`

Meldet die Session ab

Antwort `200 OK`

`POST /api/auth/pin`

Änder die Anmeldepin

Anfrage-Body

```
{ "new_pin": "987654" }
```

Antwort `200 OK`

Devices

`GET /api/devices/pending`

Gibt eine Liste der wartenden Geräte zurück

Antwort `200 OK`


```
[
  {
    "id": 1,
    "ip": "192.168.1.10",
    "first_seen": "2026-04-27T15:02:44.310Z",
    "last_seen": "2026-04-27T15:02:44.310Z",
    "request_count": 5,
    "app_name": "amp",
    "device_type": "remote",
    "discover_checked": true
  }
]
```


Feld	Typ	Beschreibung
`id`	Integer	Eindeutige ID des Eintrags
`ip`	String	IP des Gerätes
`first_seen`	TimeStamp	Wann hat sich das Gerät zum ersten mal gemeldet
`last_seen`	TimeStamp	Wann hat sich das Gerät zum letzten mal gemeldet
`request_count`	Integer	Wie oft hat sich das Gerät gemeldet

Feld	Typ	Beschreibung
`app_name`	String	Info aus dem /discover feld App (Wenn vorhanden)
`device_type`	String	Info aus dem /discover feld Type (Wenn vorhanden)
`discover_checked`	Boolean	Gibt an, ob /discover geprüft werden konnte

`POST /api/devices/pending/{device_id}/approve`

Gibt ein Gerät anhand der device_id frei, sodass dieses die Ampel steuern kann.

Abfrageparameter

Parameter	Typ	Pflicht	Beschreibung
`device_id`	Integer	Ja	Device ID

Antwort `200 OK`

`DELETE /api/devices/pending/{device_id}`

Weist ein wartendes Gerät zurück anhand der device_id.

Abfrageparameter

Parameter	Typ	Pflicht	Beschreibung
`device_id`	Integer	Ja	Device ID

Antwort `200 OK`

`GET /api/devices/approved`

Gibt eine Liste der zugelassenen Geräte zurück

Antwort `200 OK`

```
[
  {
```

```
    "id": 1,
    "ip": "192.168.1.10",
    "app_name": "amp",
    "device_type": "remote",
    "approved_at": "2026-04-27T15:13:04.210Z",
    "expires_at": "2026-04-27T15:13:04.210Z"
  }
]
```

Feld	Typ	Beschreibung
`id`	Integer	Eindeutige ID des Eintrags
`ip`	String	IP des Gerätes
`app_name`	String	Info aus dem /discover feld App (Wenn vorhanden)
`device_type`	String	Info aus dem /discover feld Type (Wenn vorhanden)
`approved_at`	TimeStamp	Gibt an, wann das Gerät zugelassen wurde
`expires_at`	TimeStamp	Gibt an, ab wann das Gerät nicht mehr zugelassen ist

`DELETE /api/devices/approved/{device_id}`

Weist ein zugelassenes Gerät zurück anhand der device_id. Dannach ist das Gerät nicht mehr berechtigt Anfragen an `/api/cmd` zu senden

Abfrageparameter

Parameter	Typ	Pflicht	Beschreibung
`device_id`	Integer	Ja	Device ID

Antwort `200 OK`

Antwortformate

Erfolg / Fehler (`DefaultStatusOUT`)

Viele Endpunkte antworten bei Erfolg oder Fehler mit folgendem Format:

```
{ "status": "ok" }
```

Hinweise

> **Wichtig:** Im Netzwerk darf nur eine Steuerung gleichzeitig laufen, da sonst die Anzeigegeräte widersprüchliche Informationen erhalten könnten.

- Alle Zeitangaben im Format ISO 8601 (`YYYY-MM-DDTHH:MM:SS`)
- Die API ist nur durch eine PIN Authentifizierung gesichert – sie sollte nur in einem gesicherten, lokalen Netzwerk betrieben werden
- Basis-URL: `http://<ip-adresse>/api/`

Revision #4
Created 2026-03-13 12:09:23 UTC by Gero Gras
Updated 2026-05-15 12:52:44 UTC by Gero Gras