

Administratoren-Handbuch

Konfiguration und Verwaltung

EINFO – Einsatzinformationssystem

Benutzerhandbuch

1. Übersicht

Das Admin-Panel ist die zentrale Verwaltungsoberfläche von EINFO. Es ist ausschließlich für Benutzer mit der Rolle „Admin“ zugänglich und erreichbar unter /user-admin.

Im Admin-Panel können Sie folgende Bereiche konfigurieren:

- Master-Key Verwaltung (Erststart und Entsperrung)
- Rollen und Berechtigungen für alle drei Boards
- Benutzerverwaltung (Anlegen, Bearbeiten, Löschen)
- Import-Einstellungen (Auto-Import und Demomodus)
- Auto-Druck für Protokolleinträge
- KI-Analyse (Situationsanalyse mit optionalem RAG-Kontext)
- Zeitgesteuerter Mailversand
- Zeitgesteuerte API-Calls
- Fetcher-Zugangsdaten
- Chatbot & Worker-Steuerung
- Knowledge-Basis (RAG) für den Chatbot
- Hybrid-Filtersystem (Regeln R1-R5)
- KI-Modell-Verwaltung (Ollama)

2. Wichtige URLs

Die folgenden Seiten sind über den Browser erreichbar (Standard-Port: 4040):

- / - Einsatzboard (Hauptansicht)
- /aufgaben - Aufgabenboard
- /status - Statusseite (druckfreundlich mit ?print=1)
- /user-login - Login-Seite
- /user-admin - Admin-Panel
- /user-firststart - Erststart-Assistent
- /Hilfe.pdf - Benutzerhandbuch Einsatzboard
- /Hilfe_Aufgabenboard.pdf - Benutzerhandbuch Aufgabenboard
- /Hilfe_Meldestelle.pdf - Benutzerhandbuch Meldestelle

3. Erststart & Master-Key

3.1 Erststart

Beim allerersten Start der Anwendung muss der Master-Key gesetzt und ein erster Admin-Benutzer angelegt werden. Navigieren Sie dazu zu /user-firststart.

- Master-Key - Wählen Sie ein sicheres Passwort als Master-Key.
- Admin-Benutzer - Benutzername und Passwort für den ersten Administrator.

Der Master-Key wird benötigt, um nach jedem Server-Neustart das System zu entsperren.

3.2 Master entsperren (nach Neustart)

Nach einem Server-Neustart ist das System gesperrt (423 Master-Lock). Navigieren Sie zum Admin-Panel (/user-admin) und geben Sie den Master-Key im Bereich „Master entsperren“ ein. Erst danach können Benutzer und Rollen verwaltet werden.

3.3 Board zurücksetzen

Im Admin-Panel steht oben rechts die Schaltfläche „Reset“ zur Verfügung. Damit wird das Einsatzboard komplett zurückgesetzt. Es erscheint eine Sicherheitsabfrage. Verwenden Sie diese Funktion nur im Notfall oder für Testszenarien.

4. Rollen und Berechtigungen

4.1 Rollenkonzept

Jede Rolle definiert die Zugriffsrechte auf die drei Boards: Einsatzboard, Aufgabenboard und Protokoll (Meldestelle). Pro Board gibt es drei Berechtigungsstufen:

- none – Kein Zugriff auf dieses Board.
- view – Nur-Ansicht. Der Benutzer kann Daten sehen, aber nicht ändern.
- edit – Vollzugriff. Der Benutzer kann anlegen, bearbeiten und löschen.

Die Rolle „Admin“ hat immer „edit“ auf allen Boards und kann nicht gelöscht oder eingeschränkt werden.

4.2 Standard-Rollen

Rolle	Beschreibung	Berechtigung
Admin	Systemadministrator	edit auf allen Boards
LtStb	Leiter Stab	edit auf allen Boards
S1	Personal	view/edit je nach Config
S2	Lage und Information	edit auf allen Boards
S3	Einsatz / Operation	view/edit je nach Config
S4	Versorgung / Logistik	view/edit je nach Config
S5	Öffentlichkeitsarbeit	view/edit je nach Config
S6	IT / Kommunikation	view/edit je nach Config
MS	Meldestelle	edit auf Protokoll
Mitarbeiter	Allgemeiner Mitarbeiter	view

4.3 Rollen verwalten

Im Bereich „Rollen (Admin + weitere)“ können Sie:

- Neue Rollen hinzufügen: Name eingeben und „Hinzufügen“ klicken.
- Rollen entfernen: Auf das × neben dem Rollennamen klicken.
- Rechte pro Rolle: In der Tabelle „Rechte pro Rolle“ die Berechtigungsstufe (none/view/edit) für jedes Board per Dropdown einstellen.
- Mit „Rollen speichern“ bzw. „Rechte speichern“ die Änderungen sichern.

5. Benutzerverwaltung

5.1 Benutzer anlegen

Geben Sie im Formular folgende Felder ein:

- Username – Eindeutiger Benutzername zum Einloggen.
- Passwort – Initiales Passwort für den Benutzer.
- Anzeigename – Wird in der Oberfläche angezeigt.
- Rollen – Wählen Sie eine oder mehrere Rollen aus der Liste. Mehrfachauswahl über Strg (Windows) oder ⌘ (macOS).

5.2 Benutzer bearbeiten

Klicken Sie auf „Edit“ neben einem Benutzer, um Anzeigename, Rollen oder Passwort zu ändern. Das Passwort wird nur aktualisiert, wenn ein neues eingegeben wird. Speichern Sie mit „Save“ oder brechen Sie mit „Cancel“ ab.

5.3 Benutzer löschen

Klicken Sie auf „Del“ neben einem Benutzer. Es erscheint eine Sicherheitsabfrage. Gelöschte Benutzer können nicht wiederhergestellt werden.

6. Relevante Speicherorte

Alle persistenten Daten liegen unter server/data/:

- Aufg_board_<ROLLE>.json – Board-Daten pro Rolle (z. B. Aufg_board_S2.json)
- Aufg_log.csv – Globales Aufgaben-Log
- Aufg_log_<ROLLE>.csv – Rollenbezogene Logs
- User_roles.json – Rollendefinitionen und Berechtigungen
- User_users.enc.json – Verschlüsselte Benutzerdaten
- User_authIndex.json – Login-Index
- User_master.json – Master-Key Information
- protocol.json / protocol.csv – Protokolldaten
- prints/protokoll_*.pdf – Gedruckte Protokolle
- conf/filtering_rules.json – Filterregel-Definitionen (R1-R5)
- conf/ai-analysis.json – KI-Analyse-Konfiguration
- llm_feedback/learned_filters.json – Gelernte Filtergewichte
- scenario_config.json – Szenario-Konfiguration

Der Frontend-Build (inkl. Hilfe-PDFs) liegt unter server/dist/.

7. Import-Einstellungen

7.1 Auto-Import

Der Auto-Import ruft in konfigurierbaren Intervallen externe Einsatzdaten ab. Im Admin-Panel können Sie folgende Parameter einstellen:

- Aktiviert/Deaktiviert – Schaltet den automatischen Import ein oder aus.
- Intervall (Sekunden) – Abstand zwischen zwei Import-Zyklen. Minimum: 5 Sekunden, Maximum: 3600 Sekunden (1 Stunde).
- Demomodus – Wenn aktiviert, wird der Fetcher beim Import nicht gestartet. Nützlich für Tests oder Präsentationen mit statischen Daten.

7.2 Fetcher-Zugangsdaten

Im Bereich „Fetcher-Zugangsdaten (global)“ können die Zugangsdaten für externe Datenquellen hinterlegt werden. Diese werden vom Import-Modul verwendet, um Einsatzdaten abzurufen.

8. Auto-Druck (Protokoll)

Der Auto-Druck generiert in regelmäßigen Abständen automatisch PDF-Ausdrucke der Protokolleinträge. Die Konfiguration umfasst:

- Aktiviert/Deaktiviert – Schaltet den automatischen Druck ein oder aus.
- Intervall (Minuten) – Zeitabstand zwischen zwei Druckläufen. Minimum: 1 Minute.
- Umfang (Scope) – Bestimmt, welche Einträge gedruckt werden:
 - „Intervall“ – Nur Einträge seit dem letzten Drucklauf.
 - „Alle“ – Alle vorhandenen Protokolleinträge.

Der Zeitpunkt des letzten Drucklaufs wird im Admin-Panel angezeigt. Gedruckte PDFs werden unter server/data/prints/ abgelegt.

9. KI-Analyse (Situationsanalyse)

Die KI-Analyse erstellt in regelmäßigen Abständen eine automatische Situationseinschätzung auf Basis der aktuellen Einsatz- und Protokolldaten.

- Aktiviert/Deaktiviert – Schaltet die automatische Analyse ein oder aus.
- Intervall (Minuten) – Zeitabstand zwischen zwei Analyseläufen. Wert 0 bedeutet: nur manuelle Auslösung.
- RAG-Kontext verwenden – Wenn aktiviert, werden zusätzlich Informationen aus der Knowledge-Basis (Wissensdatenbank) in die Analyse einbezogen. Dies kann die Qualität der Einschätzung verbessern, erhöht aber die Verarbeitungszeit.

10. Zeitgesteuerter Mailversand

Im Bereich „Zeitgesteuerter Mailversand“ können Sie wiederkehrende E-Mail-Versandaufträge konfigurieren. Jeder Zeitplan hat folgende Felder:

- Bezeichnung – Interner Name für den Zeitplan.
- Empfänger (An) – E-Mail-Adresse(n) der Empfänger.
- Betreff – Betreffzeile der E-Mail.
- Text – Nachrichteninhalt.
- Anhang-Pfad – Optionaler Dateipfad für einen Anhang.
- Modus – „Intervall“ (alle X Minuten) oder „Feste Uhrzeit“ (täglich zu einer bestimmten Uhrzeit).
- Aktiviert – Ob der Zeitplan aktiv ist.

Bestehende Zeitpläne können bearbeitet, gelöscht oder der letzte Versandzeitpunkt zurückgesetzt werden.

11. Zeitgesteuerte API-Calls

Im Bereich „Zeitgesteuerte API-Calls“ können automatische HTTP-Anfragen an externe Systeme konfiguriert werden. Jeder Zeitplan umfasst:

- Bezeichnung – Interner Name für den Zeitplan.
- URL – Ziel-URL für den HTTP-Aufruf.
- Methode – HTTP-Methode (GET, POST, PUT, DELETE).
- Body – Optionaler Request-Body (für POST/PUT).
- Modus – „Intervall“ (alle X Minuten) oder „Feste Uhrzeit“ (täglich).
- Aktiviert – Ob der Zeitplan aktiv ist.

Zeitpläne können bearbeitet, gelöscht oder der letzte Aufrufzeitpunkt zurückgesetzt werden.

12. Chatbot & Worker

12.1 Chatbot-Steuerung

Der EINFO-Chatbot basiert auf einem lokalen LLM (Llama 3.1) und nutzt RAG (Retrieval-Augmented Generation) für kontextbezogene Antworten. Im Admin-Panel können Sie den Chatbot starten und stoppen. Der aktuelle Status (Running/Stopped) wird automatisch alle 5 Sekunden aktualisiert.

12.2 Worker-Steuerung

Der Worker ist ein Hintergrundprozess, der regelmäßig Aufgaben wie Datenaufbereitung, Analyse und Synchronisation durchführt. Sie können den Worker starten und stoppen.

12.3 Worker-Intervall

Im Bereich „Worker-Intervall Einstellung“ legen Sie fest, wie oft der Worker seine Aufgaben ausführt. Das Intervall wird in Sekunden angegeben (Minimum: 5 Sekunden). Zusätzlich kann der Worker hier aktiviert oder deaktiviert werden.

13. Knowledge-Basis (RAG)

Die Knowledge-Basis enthält Dokumente, die der Chatbot als Wissensquelle nutzt. Neue Dateien (PDF, JSON, TXT) können hochgeladen werden.

- Dateien hochladen – Wählen Sie eine oder mehrere Dateien über den Upload-Button aus.
- Dateien anzeigen – Die Liste zeigt alle vorhandenen Dateien in der Knowledge-Basis mit Dateiname und Größe.
- Dateien löschen – Einzelne Dateien können aus der Knowledge-Basis entfernt werden.
- Ingest starten – Nach dem Hochladen neuer Dateien muss ein Ingest (Indizierung) gestartet werden, damit die Inhalte im RAG-System verfügbar werden. Dieser Vorgang kann einige Minuten dauern.

14. Hybrid-Filtersystem (R1-R5)

Das Filtersystem besteht aus fünf konfigurierbaren Regeln, die steuern, welche Daten dem Chatbot als Kontext bereitgestellt werden. Die Regeln können einzeln aktiviert oder deaktiviert werden.

R1 - Abschnitte-Priorität

Filtert Abschnitte nach Priorität und zeigt die wichtigsten. Berücksichtigt kritische Einsätze, Gesamtzahl der Einsätze, Personalstärke und durchschnittlichen Personaleinsatz pro Einsatz.

R2 - Protokoll-Relevanz

Filtert Protokoll-Einträge nach Relevanz. Bewertet Einträge anhand konfigurierbarer Faktoren wie offene Fragen, Ressourcen-Anfragen, Statusmeldungen, Dringlichkeit und Warnungen. Einige Faktoren sind „lernbar“ und passen ihre Gewichtung automatisch an.

R3 - Trend-Erkennung

Erkennt Trends in der Einsatzentwicklung über konfigurierbare Zeitfenster (Standard: 60 und 120 Minuten). Erstellt Prognosen für den zukünftigen Einsatzverlauf.

R4 - Ressourcen-Status

Analysiert den Ressourcen-Status und erkennt Engpässe. Hebt Bereiche mit hoher Auslastung hervor (Standard-Schwelle: 80%).

R5 - Stabs-Fokus

Aggregiert Daten für die Stabs-Ansicht. Zeigt nur kritische Einzeleinsätze (z.B. Personen in Gefahr, Evakuierungen, kritische Infrastruktur). Die Scoring-Faktoren und Schwellenwerte sind im Admin-Panel konfigurierbar.

Gelernte Filter

Das System lernt aus Benutzer-Feedback automatisch, welche Filterkriterien hilfreich waren. Die gelernten Gewichte können im Admin-Panel eingesehen und bei Bedarf zurückgesetzt werden.

15. KI-Modell-Verwaltung

Im Bereich „KI-Modell-Verwaltung“ werden die lokal verfügbaren LLM-Modelle (via Ollama) verwaltet. Sie können:

- Verfügbare Modelle auflisten und deren Status einsehen.
- Neue Modelle herunterladen (Pull).
- Das aktive Modell für den Chatbot und die Analyse auswählen.

16. Konfiguration (.env)

Alle Umgebungsvariablen werden in der Datei server/.env (bzw. .env) konfiguriert. Nach Änderungen muss der Server neu gestartet werden. Im Folgenden sind alle verfügbaren Parameter dokumentiert.

16.1 Server-Grundkonfiguration

- PORT – HTTP-Port des Hauptservers (Standard: 4040).
- DATA_DIR – Basisverzeichnis für persistente Daten (Einsatzlisten, Sessions, Druckausgaben).
- PUBLIC_DIR – Optionales Verzeichnis für statische WMS-Dateien.
- KANBAN_LOG_DIR – Optionales Log-Verzeichnis (logs/Log.txt, WMS_TILES.log).
- KANBAN_COOKIE_SECURE – Auf "1" setzen, um sichere Cookies für Board-Login zu erzwingen (HTTPS erforderlich).

16.2 Frontend-Polling

- UI_STATUS_POLL_INTERVAL_MS – Polling-Intervall (ms) für /api/ff/status und /api/ff/creds.
- UI_ACTIVITY_POLL_INTERVAL_MS – Polling-Intervall (ms) für /api/activity/status.

16.3 Cache-Konfiguration

- BOARD_CACHE_MAX_AGE_MS – Maximale Cache-Dauer für berechnete Board-Daten in Millisekunden.
- VEHICLE_CACHE_TTL_MS – Lebensdauer des Fahrzeug-Caches bevor er neu geladen wird (ms).

16.4 Feuerwehr-Feed (Fetcher)

- FF_OUT_FILE – Ausgabedatei fuer den gefilterten Feed (JSON-Format).
- FF_GPS_OUT_FILE – Ausgabedatei fuer Fahrzeug-GPS-Informationen.
- FF_POLL_INTERVAL_MS – Poll-Intervall fuer den Feed (Standard: 60000 ms = 1 Minute).
- FF_ACTIVITY_SWEEP_INTERVAL_MS – Intervall fuer die Aktivitaetsueberwachung (ms).
- FF_DEBUG – Auf "1" setzen fuer detailliertes Fetcher-Logging (HTTP-Status, Parsing-Infos).
- FF_LIST_PATH – Pfadsegment fuer den Einsatzlisten-Endpunkt (Standard: "/list").
- FF_LIST_EXTRA – Zusaetzliche Query-Parameter fuer den Einsatzlisten-Endpunkt.
- FF_LIST_TIMEOUT_MIN – Maximales Timeout in Minuten, bevor der Feed als veraltet markiert wird (Standard: 2880).
- FF_GPS_PATH – Pfadsegment fuer den Fahrzeug-GPS-Endpunkt (Standard: "/status/gps").
- FF_ONCE – "1" fuer einmaligen Abruf (Debug/Test), "0" fuer Dauerbetrieb.
- FF_CA_FILE – Optionaler Pfad zur TLS-Zertifikatskette fuer HTTPS-Verbindungen.
- FF_LOCK_FILE – Optionaler Pfad zur Lock-Datei, um parallele Fetcher-Instanzen zu verhindern.
- FF_AUTO_STOP_MIN – Minuten bis zur automatischen Abschaltung eines Einsatzes ohne neue Ereignisse (optional).
- FF_USERNAME – Optionaler HTTP-Basic-Auth-Benutzername fuer den Feed-Zugriff.
- FF_PASSWORD – Optionales Passwort / API-Secret fuer den Feed-Zugriff.
- FF_LOGIN_MAX_RETRIES – Maximale Login-Wiederholungen (Standard: 3).
- FF_LOGIN_RETRY_DELAY_MS – Verzoegerung zwischen Login-Versuchen in ms (Standard: 5000).

16.5 Naehe-Suche (Nearby)

- NEARBY_RADIUS_KM – Standard-Suchradius in km (Standard: 10).
- NEARBY_RADIUS_MIN_KM – Minimaler Radius in km (Standard: 0.1).
- NEARBY_RADIUS_MAX_KM – Maximaler Radius in km (Standard: 50).

16.6 WMS / Karten-Konfiguration

- WMS_PORT – Port des WMS-Dienstes (Standard: 8090).
- WMS_TITLE – Titel fuer die WMS-Capabilities-Metadaten.
- WMS_ABSTRACT – Beschreibung fuer WMS-Metadaten.
- WMS_LABELS – "1" zeigt Kartenbeschriftungen, "0" nur Symbole.
- WMS_LABEL_FONT – CSS-Fontangabe fuer Beschriftungen (Standard: "12px Sans-Serif").
- WMS_LABEL_COLOR – Hex-Farbe fuer Beschriftungstext (Standard: "#000000").
- WMS_LABEL_OUTLINE – Hex-Farbe fuer Beschriftungsumriss (Standard: "#ffffff").
- WMS_LABEL_OUTLINE_W – Breite des Umriss-Strichs in Pixeln (Standard: 3).
- WMS_LABEL_TRIM – Maximale Textlaenge bevor abgeschnitten wird (Standard: 28).
- WMS_DEBUG – "1" aktiviert WMS-Debug-Logging.

16.7 Druck-Konfiguration

- KANBAN_MELDUNG_PRINT_DIR – Ausgabeverzeichnis fuer Meldungsdrucke.
- KANBAN_EINSATZ_PRINT_DIR – Ausgabeverzeichnis fuer Einsatzdrucke.
- KANBAN_PROTOKOLL_PRINT_DIR – Ausgabeverzeichnis fuer Protokolldrucke.
- KANBAN_PRINT_COMMAND – Druckbefehl (Standard: "lp").
- KANBAN_PRINT_OUTPUT_DIR – Explizites Druckausgabeverzeichnis.
- PRINT_BASE_DIR – Basisverzeichnis fuer Druckausgaben.
- PUPPETEER_EXECUTABLE_PATH – Benutzerdefinierter Pfad zur Chrome/Chromium-Binary fuer die PDF-Erzeugung.

16.8 Benutzer-Sessions & Online-Status

- USER_SESSION_IDLE_TIMEOUT_MIN – Inaktivitäts-Timeout in Minuten (Standard: 15).
- USER_SESSION_IDLE_TIMEOUT_MS – Timeout in Millisekunden (Standard: 900000).
- USER_SESSION_SWEEP_INTERVAL_MS – Intervall für die Bereinigung inaktiver Sessions in ms (Standard: 60000).

16.9 Aufgaben / Frist-Konfiguration

- DEFAULT_DUE_OFFSET_MINUTES – Standard-Vorlaufzeit für Aufgabenfristen in Minuten (Standard: 10).
- TASK_DEFAULT_DUE_OFFSET_MINUTES – Alias für obigen Wert.
- AUFG_DEFAULT_DUE_MINUTES – Deutscher Alias für obigen Wert.

16.10 Auto-Import & Auto-Druck

- AUTO_IMPORT_DEFAULT_INTERVAL_SEC – Standard-Auto-Import-Intervall in Sekunden (Standard: 30).
- AUTO_PRINT_DEFAULT_INTERVAL_MINUTES – Standard-Auto-Druck-Intervall in Minuten (Standard: 10).
- AUTO_PRINT_MIN_INTERVAL_MINUTES – Minimales Auto-Druck-Intervall in Minuten (Standard: 1).

16.11 Mail / SMTP-Konfiguration

- MAIL_HOST – SMTP-Server-Hostname.
- MAIL_PORT – SMTP-Port (Standard: 587 fuer STARTTLS, 465 fuer SMTPTS, 25 fuer unverschlueselt).
- MAIL_SECURE – "1" fuer SMTPTS (TLS ab Verbindungsstart), "0" fuer STARTTLS.
- MAIL_STARTTLS – "1" um STARTTLS nach dem Verbindungsaufbau anzufordern.
- MAIL_USER / MAIL_USERNAME – SMTP-Benutzername.
- MAIL_PASSWORD / MAIL_PASS – SMTP-Passwort.
- MAIL_FROM – Absender-E-Mail-Adresse.
- MAIL_REPLY_TO – Antwortadresse (Reply-To).
- MAIL_ALLOWED_FROM – Kommagetrennte Liste erlaubter Absenderadressen.
- MAIL_TIMEOUT_MS – SMTP-Befehls-Timeout in ms (Standard: 15000).
- MAIL_TLS_REJECT_UNAUTHORIZED – "1" um selbstsignierte Zertifikate abzulehnen.
- MAIL_CLIENT_ID – EHLO/HELO-Kennung.
- MAIL_LOG – "1" aktiviert detailliertes Mail-Logging.
- MAIL_DELETE_AFTER_READ – "1" um Mails nach Verarbeitung zu loeschen (Standard: 1).
- MAIL_INBOX_DIR – Benutzerdefinierter Inbox-Verzeichnispfad.
- MAIL_INBOX_POLL_INTERVAL_SEC – Mail-Prueflintervall in Sekunden.

16.12 IMAP-Konfiguration

- MAIL_IMAP_HOST – IMAP-Server-Hostname.
- MAIL_IMAP_PORT – IMAP-Port (Standard: 993 fuer TLS, 143 fuer Klartext/StartTLS).
- MAIL_IMAP_SECURE – "1" fuer TLS ab Verbindungsstart, "0" fuer Klartext.
- MAIL_IMAP_USER – IMAP-Benutzername.
- MAIL_IMAP_PASSWORD – IMAP-Passwort.
- MAIL_IMAP_MAILBOX – Postfach-Ordnername (Standard: "INBOX").
- MAIL_IMAP_TLS_REJECT_UNAUTHORIZED – "1" um selbstsignierte Zertifikate abzulehnen.

16.13 POP3-Konfiguration

- MAIL_POP3_HOST – POP3-Server-Hostname.
- MAIL_POP3_PORT – POP3-Port (Standard: 995 fuer TLS, 110 fuer Klartext).
- MAIL_POP3_SECURE – "1" fuer TLS ab Verbindungsstart, "0" fuer Klartext.
- MAIL_POP3_USER – POP3-Benutzername.
- MAIL_POP3_PASSWORD – POP3-Passwort.
- MAIL_POP3_TLS_REJECT_UNAUTHORIZED – "1" um selbstsignierte Zertifikate abzulehnen.

16.14 Chatbot-Integration

- CHATBOT_BASE_URL – URL des Chatbot-Servers (Standard: "http://127.0.0.1:3100").
- CHATBOT_PORT – Chatbot-Server-Port (Standard: 3100).
- CHATBOT_HOST – Chatbot-Bind-Adresse (Standard: "0.0.0.0").
- CHATBOT_PROFILE – Konfigurationsprofil-Name (Standard: "default").
- CHATBOT_DEBUG – "1" aktiviert Chatbot-Debug-Logging.
- CHATBOT_AUTO_STEP_MS – Auto-Step-Intervall in ms (Standard: 120000).
- DEBUG_PROXY – "1" aktiviert Proxy-Debugging.
- DEBUG_SITUATION – "1" aktiviert Situations-Debugging.

16.15 LLM-Konfiguration

- LLM_BASE_URL – Ollama/LLM-Server-URL (Standard: "http://127.0.0.1:11434").
- LLM_CHAT_MODEL – Chat-Modellname (Standard: "llama3.1:8b").
- LLM_EMBED_MODEL – Embedding-Modellname (Standard: "mxbai-embed-large").
- LLM_TEMP – Standard-Temperatur (Standard: 0.05).
- LLM_SEED – Zufallswert / Seed (Standard: 42).
- LLM_CHAT_TIMEOUT_MS – Chat-Request-Timeout in ms (Standard: 60000).
- LLM_SIM_TIMEOUT_MS – Simulations-Timeout in ms (Standard: 300000).
- LLM_EMBED_TIMEOUT_MS – Embedding-Timeout in ms (Standard: 30000).
- LLM_TIMEOUT_MS – Allgemeines Timeout in ms (Standard: 240000).
- LLM_NUM_CTX – Kontext-Fenstergroesse (Standard: 8192).
- LLM_NUM_BATCH – Batch-Groesse (Standard: 512).

16.16 GPU / Ollama

- CUDA_VISIBLE_DEVICES – Zu verwendende GPU-Geraete (Standard: "0").
- OLLAMA_NUM_GPU – Anzahl der GPU-Layer (Standard: 22 fuer 8 GB VRAM auf RTX 4070).
- OLLAMA_MAX_LOADED_MODELS – Max. gleichzeitig geladene Modelle (Standard: 1).
- OLLAMA_KEEP_ALIVE – Keep-Alive-Dauer fuer Modelle (Standard: "30m").

16.17 RAG-Konfiguration

- RAG_DIM – Embedding-Dimension (Standard: 1024).
- RAG_TOP_K – Anzahl Top-K-Ergebnisse (Standard: 10).
- RAG_MAX_CTX – Maximale Kontext-Zeichen (Standard: 4000).
- RAG_MAX_ELEM – Maximale Index-Elemente (Standard: 50000).
- RAG_SCORE_THRESHOLD – Score-Schwellenwert (Standard: 0.2).
- EMBED_CACHE_SIZE – Embedding-Cache-Groesse (Standard: 200).

16.18 Prompt-Konfiguration

- PROMPT_MAX_BOARD – Max. Board-Eintraege im Prompt (Standard: 25).
- PROMPT_MAX_AUFGABEN – Max. Aufgaben-Eintraege (Standard: 50).
- PROMPT_MAX_PROTOKOLL – Max. Protokoll-Eintraege (Standard: 30).

16.19 Memory / RAG-Langzeit

- MEM_RAG_LONG_MIN_ITEMS – Min. Eintraege fuer Langzeit-Szenario (Standard: 100).
- MEM_RAG_MAX_AGE_MIN – Max. Alter in Minuten (Standard: 720).
- MEM_RAG_HALF_LIFE_MIN – Halbwertszeit fuer Aktualitaet in Minuten (Standard: 120).
- MEM_RAG_LONG_TOP_K – Top-K fuer Langzeit-Szenario (Standard: 12).

16.20 Simulation

- SIM_WORKER_INTERVAL_MS – Simulations-Worker-Intervall in ms (Standard: 60000).
- SIM_MAX_RETRIES – Max. Wiederholungen (Standard: 3).
- SIM_RETRY_DELAY_MS – Verzoegerung zwischen Wiederholungen in ms (Standard: 5000).
- MAIN_SERVER_URL – URL des Hauptservers (Standard: "http://localhost:4000").

16.21 Experimentelle Features

- EINFO_EXPERIMENTAL_SCENARIOPACK – "1" aktiviert experimentelle Szenariopakete (Standard: 0).
- EINFO_DATA_DIR – Alternatives Datenverzeichnis fuer den Chatbot.

16.22 Sonstige

- WEATHER_WARNING_DATE_FILE – Pfad fuer die Wetterwarnungs-Datumsdatei.

16.23 Client-seitige Umgebungsvariablen (Vite)

Diese Variablen werden zur Build-Zeit ausgewertet und muessen mit dem Praefix VITE_ beginnen:

- VITE_API_BASE_URL – URL des API-Servers.
- VITE_LOGIN_BASE_URL – URL des Login-Servers.
- VITE_CHATBOT_BASE_URL – URL des Chatbot-Servers.
- VITE_STATUS_POLL_INTERVAL_MS – Status-Polling-Intervall in ms (Standard: 3000).
- VITE_ACTIVITY_POLL_INTERVAL_MS – Aktivitaets-Polling-Intervall in ms (Standard: 1000).

17. API-Referenz

Alle API-Endpunkte sind unter /api/ erreichbar und erfordern (sofern nicht anders angegeben) eine gueltige Benutzer-Session. Antworten erfolgen im JSON-Format.

17.1 Einsatzboard & Fahrzeuge

- GET /api/board – Gibt das komplette Einsatzboard mit allen Spalten zurueck.
- GET /api/vehicles – Listet alle Fahrzeuge (Basis + Zusatz) auf.
- POST /api/vehicles – Legt ein neues Fahrzeug an oder klonnt ein bestehendes.
- PATCH /api/vehicles/:id/availability – Aendert die Verfuegbarkeit eines Fahrzeugs.
- PATCH /api/vehicles/:id/position – Aktualisiert die GPS-Position eines Fahrzeugs.
- DELETE /api/vehicles/:id/position – Loescht eine manuelle Positions-Ueberschreibung.
- GET /api/groups/availability – Gibt den Verfuegbarkeitsstatus aller Gruppen zurueck.
- GET /api/groups/alerted – Listet alarmierte Gruppen auf.
- PATCH /api/groups/:name/availability – Aendert die Verfuegbarkeit einer Gruppe.
- GET /api/gps – Gibt GPS-Daten zurueck.
- GET /api/types – Gibt verfuegbare Einsatztypen zurueck.
- GET /api/nearby – Sucht naechstgelegene Einheiten/Ressourcen.

17.2 Einsatzkarten (Cards)

- POST /api/cards – Legt eine neue Einsatzkarte an.
- POST /api/cards/:id/move – Verschiebt eine Karte in eine andere Spalte.
- POST /api/cards/:id/assign – Weist ein Fahrzeug einer Karte zu.
- POST /api/cards/:id/unassign – Entfernt ein Fahrzeug von einer Karte.
- PATCH /api/cards/:id/personnel – Aktualisiert die Personalstaerke einer Karte.
- PATCH /api/cards/:id – Aktualisiert Karten-Eigenschaften (Titel, Ort, Typ, Koordinaten, Abschnitt usw.).

17.3 Protokoll (Meldestelle)

- GET /api/protocol – Listet alle Protokolleinträge auf.
- POST /api/protocol – Erstellt einen neuen Protokolleintrag.
- GET /api/protocol/:nr – Gibt einen bestimmten Eintrag zurück.
- PUT /api/protocol/:nr – Aktualisiert einen Protokolleintrag.
- POST /api/protocol/:nr/lock – Sperrt einen Eintrag zur Bearbeitung.
- DELETE /api/protocol/:nr/lock – Gibt die Sperre eines Eintrags frei.
- GET /api/protocol/csv/file – Exportiert das Protokoll als CSV-Datei.
- GET /api/protocol/auto-print-config – Gibt die Auto-Druck-Konfiguration zurück (nur Admin).
- POST /api/protocol/auto-print-config – Aktualisiert die Auto-Druck-Konfiguration (nur Admin).

17.4 Aufgaben

- GET /api/aufgaben – Listet alle Aufgaben für die aktuelle Rolle auf.
- POST /api/aufgaben – Erstellt eine neue Aufgabe.
- POST /api/aufgaben/:id/edit – Bearbeitet eine bestehende Aufgabe.
- POST /api/aufgaben/:id/status – Ändert den Status einer Aufgabe.
- POST /api/aufgaben/reorder – Sortiert Aufgaben um.
- GET /api/aufgaben/config – Gibt die Aufgabenkonfiguration zurück.
- GET /api/aufgaben/protocols – Gibt Protokolle für die aktuelle Rolle zurück.

17.5 Mail

- GET /api/mail/status – Gibt den Mail-Konfigurationsstatus zurück.
- POST /api/mail/send – Versendet eine E-Mail.
- GET /api/mail/inbox/status – Prüft den Inbox-Status.
- GET /api/mail/inbox – Listet Inbox-Nachrichten auf (mit optionalem limit-Parameter).
- GET /api/mail/schedule – Listet alle Mail-Zeitpläne auf (nur Admin).
- POST /api/mail/schedule – Erstellt einen neuen Mail-Zeitplan (nur Admin).
- PUT /api/mail/schedule/:id – Aktualisiert einen Mail-Zeitplan (nur Admin).
- DELETE /api/mail/schedule/:id – Löscht einen Mail-Zeitplan (nur Admin).

17.6 HTTP-API-Zeitplaene

- GET /api/http/schedule – Listet alle HTTP-Zeitplaene auf (nur Admin).
- POST /api/http/schedule – Erstellt einen neuen HTTP-Zeitplan (nur Admin).
- PUT /api/http/schedule/:id – Aktualisiert einen HTTP-Zeitplan (nur Admin).
- DELETE /api/http/schedule/:id – Loescht einen HTTP-Zeitplan (nur Admin).

17.7 Import & Export

- GET /api/import/auto-config – Gibt die Auto-Import-Konfiguration zurueck.
- POST /api/import/auto-config – Aktualisiert die Auto-Import-Konfiguration.
- POST /api/import/trigger – Loest einen sofortigen Import aus.
- GET /api/export/pdf – Exportiert das Board als PDF.
- GET /api/log.csv – Laedt das Aktivitaetsprotokoll als CSV herunter.

17.8 Feuerwehr-Fetcher

- GET /api/ff/status – Gibt den Fetcher-Status zurueck.
- GET /api/ff/status/details – Gibt detaillierten Fetcher-Status zurueck.
- GET /api/ff/creds – Prueft, ob Fetcher-Zugangsdaten vorhanden sind.
- POST /api/ff/creds – Speichert Fetcher-Zugangsdaten.
- POST /api/ff/start – Startet den Fetcher-Dienst.
- POST /api/ff/stop – Stoppt den Fetcher-Dienst.

17.9 Drucken

- POST /api/print/server – Druckt ein PDF ueber den Systemdrucker.
- GET /api/print/server/info – Gibt Drucker-Informationen zurueck.
- POST /api/print/:nr/print – Druckt einen Protokolleintrag.
- POST /api/print/blank/print – Druckt ein leeres Protokollformular.
- GET /api/print/:nr/print/file/:file – Gibt eine gedruckte Protokolldatei zurueck.
- GET /api/print/blank/print/file/:file – Gibt eine leere Protokolldatei zurueck.
- POST /api/incident-print/:incidentId/print – Speichert einen Einsatz als PDF.
- POST /api/incident-print/:incidentId/mail – Versendet einen Einsatz per E-Mail.

17.10 KI-Analyse & Situationsanalyse

- GET /api/situation/status – Gibt den Analyse-Status zurück.
- POST /api/situation/analysis-loop-sync – Synchronisiert den Analyse-Zyklus.
- GET /api/situation/analysis – Gibt das Analyse-Ergebnis zurück.
- POST /api/situation/question – Stellt eine Situations-Frage.
- POST /api/situation/suggestion/feedback – Gibt Feedback zu einem Vorschlag.
- POST /api/situation/question/feedback – Gibt Feedback zu einer Antwort.
- GET /api/situation/analysis-config – Gibt die Analyse-Konfiguration zurück.
- POST /api/situation/analysis-config – Aktualisiert die Analyse-Konfiguration.

17.11 Admin-Filterregeln

- GET /api/admin/filtering-rules/status – Gibt den Filterstatus zurück.
- GET /api/admin/filtering-rules – Listet alle Filterregeln auf.
- PUT /api/admin/filtering-rules – Aktualisiert die Filterregeln.
- GET /api/admin/filtering-rules/learned – Gibt gelernte Filtergewichte zurück.
- POST /api/admin/filtering-rules/reset-learned – Setzt gelernte Gewichte zurück.
- GET /api/admin/filtering-rules/ai-analysis-config – Gibt die KI-Analyse-Konfiguration zurück.
- PUT /api/admin/filtering-rules/ai-analysis-config – Aktualisiert die KI-Analyse-Konfiguration.
- GET /api/admin/filtering-rules/scenario – Gibt die Szenario-Konfiguration zurück.
- PUT /api/admin/filtering-rules/scenario – Aktualisiert die Szenario-Konfiguration.

17.12 Benutzer & Rollen

- GET /api/user/roles – Gibt alle Rollen zurück.
- PUT /api/user/roles – Aktualisiert die Rollen (nur Admin).
- GET /api/user/online-roles – Gibt online-aktive Rollen zurück.

17.13 Admin-Verwaltung

- POST /api/user/admin/initialsetup – Erststart: Initialisiert das System mit Standarddaten.
- POST /api/user/admin/archive – Erstellt ein ZIP-Archiv aller Daten.
- GET /api/user/admin/archive/create-download – Erstellt ein Archiv und gibt es zum Download zurueck.
- GET /api/user/admin/archive/download/:file – Laedt eine bestimmte Archivdatei herunter.
- GET /api/user/admin/archive/testlist – Listet Test-Archive auf.
- GET /api/user/admin/logs/download – Laedt Logdateien herunter.
- GET /api/user/admin/chatbot/status – Gibt den Chatbot-Service-Status zurueck.
- POST /api/user/admin/chatbot/start – Startet den Chatbot-Service.
- POST /api/user/admin/chatbot/stop – Stoppt den Chatbot-Service.
- POST /api/user/admin/chatbot/server/start – Startet den Chatbot-Server.
- POST /api/user/admin/chatbot/server/stop – Stoppt den Chatbot-Server.
- POST /api/user/admin/chatbot/worker/start – Startet den Chatbot-Worker.
- POST /api/user/admin/chatbot/worker/stop – Stoppt den Chatbot-Worker.
- GET /api/user/admin/worker/config – Gibt die Worker-Konfiguration zurueck.
- PATCH /api/user/admin/worker/config – Aktualisiert die Worker-Konfiguration.

17.14 Knowledge-Basis (RAG)

- GET /api/user/admin/knowledge/files – Listet Knowledge-Dateien auf.
- POST /api/user/admin/knowledge/upload – Laedt eine einzelne Knowledge-Datei hoch.
- POST /api/user/admin/knowledge/upload-multiple – Laedt mehrere Dateien hoch (max. 20).
- DELETE /api/user/admin/knowledge/files/:filename – Loescht eine Knowledge-Datei.
- POST /api/user/admin/knowledge/ingest – Startet die Indizierung der Knowledge-Basis.

17.15 Aktivitaet & Status

- GET /api/activity/status – Gibt den Systemaktivitaetsstatus zurueck (oeffentlich, keine Authentifizierung erforderlich).

17.16 Chatbot-Server API (Port 3100)

Die folgenden Endpunkte sind auf dem separaten Chatbot-Server verfuegbar und werden vom Hauptserver teilweise als Proxy weitergeleitet.

Szenarien & Simulation

- GET /api/scenarios – Listet alle verfuegbaren Szenarien auf.
- GET /api/scenarios/:scenarioid – Gibt Details eines Szenarios zurueck.
- POST /api/sim/start – Startet eine Simulation (mit optionalem Szenario).
- GET /api/sim/status – Gibt den Simulations-Status zurueck.
- GET /api/sim/scenario – Gibt das aktive Szenario zurueck.
- POST /api/sim/pause – Pausiert die Simulation.
- POST /api/sim/step – Fuehrt einen einzelnen Simulationsschritt aus.
- POST /api/sim/waiting-for-roles – Signalisiert Warten auf Rollen.

Chat & LLM

- POST /api/chat – Sendet eine Chat-Nachricht (Rate-Limit: 60/min).
- GET /api/llm/models – Listet verfuegbare LLM-Modelle auf.
- GET /api/llm/gpu – Gibt den GPU-Status zurueck.
- GET /api/llm/system – Gibt den Systemstatus zurueck.
- POST /api/llm/test – Testet das LLM mit einer Frage (Rate-Limit: 10/min).
- GET /api/llm/config – Gibt die LLM-Konfiguration zurueck.
- POST /api/llm/global-model – Setzt das globale LLM-Modell.
- POST /api/llm/task-config – Konfiguriert aufgabenspezifische LLM-Einstellungen.
- GET /api/llm/model/:taskType – Gibt das Modell fuer einen Aufgabentyp zurueck.
- POST /api/llm/test-model – Testet ein bestimmtes Modell (Rate-Limit: 10/min).
- POST /api/llm/test-with-metrics – Testet mit Metriken (Rate-Limit: 10/min).
- POST /api/llm/test-with-metrics-stream – Testet mit Metriken als Stream (Rate-Limit: 10/min).
- GET /api/llm/profiles – Gibt verfuegbare LLM-Profile zurueck.
- GET /api/llm/prompt-templates – Listet Prompt-Templates auf.
- GET /api/llm/prompt-templates/:name – Gibt ein bestimmtes Prompt-Template zurueck.
- PUT /api/llm/prompt-templates/:name – Aktualisiert ein Prompt-Template (Rate-Limit: 10/min).
- GET /api/llm/action-history – Gibt die LLM-Aktionshistorie zurueck.
- GET /api/llm/ops-verworfen – Gibt verworfene Operationen zurueck.
- GET /api/llm/exchange/:exchangeld – Gibt einen bestimmten LLM-Austausch zurueck.

Metriken & Monitoring

- GET /api/metrics – Gibt Simulations-Metriken zurueck.
- GET /api/metrics/stats – Gibt Metrik-Statistiken zurueck.
- GET /api/events – Event-Stream (Server-Sent Events).

Audit

- GET /api/audit/status – Gibt den Audit-Status zurueck.
- POST /api/audit/start – Startet eine Audit-Aufzeichnung.
- POST /api/audit/end – Beendet eine Audit-Aufzeichnung.
- GET /api/audit/list – Listet alle Audit-Sessions auf.
- GET /api/audit/:exerciseld – Gibt Audit-Daten fuer eine Uebung zurueck.
- DELETE /api/audit/:exerciseld – Loescht Audit-Daten.
- POST /api/audit/pause – Pausiert die Audit-Aufzeichnung.
- POST /api/audit/resume – Setzt die Audit-Aufzeichnung fort.
- POST /api/audit/events – Zeichnet Audit-Ereignisse auf.

Templates & Uebungen

- GET /api/templates – Listet alle Templates auf.
- GET /api/templates/:templateId – Gibt ein bestimmtes Template zurueck.
- POST /api/templates – Erstellt ein neues Template.
- DELETE /api/templates/:templateId – Loescht ein Template.
- POST /api/templates/:templateId/create-exercise – Erstellt eine Uebung aus einem Template.

Katastrophen-Kontext

- GET /api/disaster/current – Gibt den aktuellen Katastrophen-Kontext zurueck.
- GET /api/disaster/summary – Gibt eine gefilterte Zusammenfassung zurueck.
- POST /api/disaster/init – Initialisiert einen Katastrophen-Kontext.
- POST /api/disaster/update – Aktualisiert den Kontext aus EINFO-Daten.
- GET /api/disaster/list – Listet alle Katastrophen auf.
- GET /api/disaster/:disasterId – Gibt eine bestimmte Katastrophe zurueck.
- POST /api/disaster/finalize – Schliesst einen Katastrophen-Kontext ab.
- POST /api/disaster/record-suggestion – Zeichnet einen Vorschlag auf.

Feedback & Lernen

- POST /api/feedback – Gibt Feedback ab.
- GET /api/feedback/list – Listet Feedback-Eintraege auf.
- GET /api/feedback/stats – Gibt Feedback-Statistiken zurueck.
- POST /api/feedback/similar – Sucht aehnliches Feedback.
- POST /api/feedback/learned-context – Zeichnet gelernten Kontext auf.

18. Backup & Recovery

Erstellen Sie regelmäßig Backups des Verzeichnisses server/data/ vor größeren Änderungen. Das Verzeichnis enthält alle persistenten Daten:

- Benutzer- und Rollendaten
- Einsatz- und Aufgabenboards
- Protokolldaten und gedruckte PDFs
- Filterregeln und gelernte Gewichte
- Szenario- und Analyse-Konfigurationen

Bei verlorenem Admin-Zugang

Sichern Sie die Dateien User_master.json und User_users.enc.json. Entfernen Sie diese Dateien und starten Sie den Server neu. Navigieren Sie zu /user-firststart, um einen neuen Master-Key und Admin-Benutzer anzulegen. Die Rollendefinitionen (User_roles.json) bleiben dabei erhalten.

Wartung

Im Admin-Panel steht im Bereich „Wartung (Admin)“ eine Funktion zum Herunterladen von Backup-Dateien und Logdateien zur Verfügung. Nutzen Sie diese regelmäßig, um Datenverluste zu vermeiden.

Anhang: URL- und Parameter-Referenz

Dieses Kapitel dokumentiert alle in EINFO definierten HTTP-Endpunkte (API und Nicht-API), inklusive Path-, Query- und relevanter Header-Parameter sowie Beispielen.

Nicht-/api

UI - Status/Print

GET /status	Query: font (0.5..5); bereichId abschnittId areald; bereich abschnitt area; print=1; pdf=1 Wirkung: Liefert die Status-Seite (SPA). Query-Parameter werden client-seitig ausgewertet (Filter/Print/Font). Hinweis: Server liefert index.html; Parameter werden im Browser (StatusPage.jsx) verarbeitet. Beispiele: <code>/status?font=1.4</code> <code>/status?bereichId=3&bereich=Abschnitt%203</code> <code>/status?pdf=1</code>
-------------	--

UI - Protokoll Prefill

GET /#/protokoll	Query: prefillToken; prefill Wirkung: Protokoll-Maske (SPA). Hash-Query triggert Prefill aus Aufgaben-Kontext. Hinweis: Client-seitig (ProtokollPage.jsx) – nicht serverseitig. Beispiele: <code>/#/protokoll?prefillToken=&lt;token&gt;</code> <code>/#/protokoll?prefill=&lt;token&gt;</code>
------------------	--

Karte - WMS/GeoJSON/Tiles

GET /geojson	Query: layers=lage:all lage:incidents lage:vehicles; crs=EPSG:4326 3857; bbox=minx,miny,maxx,maxy Wirkung: GeoJSON Features für Kartenlayer; optionaler BBox-Filter. Hinweis: WMS-lite Server. Beispiele: <code>/geojson?layers=lage:all</code> <code>/geojson?layers=lage:vehicles&crs=EPSG:4326&bbox=14.0,46.6,14.2,46.8</code>
GET /tiles/:layer/:z/:x/:y.png	Path: layer (Einsatz FZG); z; x; y Wirkung: PNG Tiles (XYZ). Hinweis: WMS-lite Server. Beispiele: <code>/tiles/Einsatz/12/2210/1412.png</code>
GET /wms	Query: service=WMS; request=GetCapabilities Wirkung: WMS-lite Capabilities. Andere request-Werte führen zu 400 mit Hinweis /tiles zu nutzen. Hinweis: Wird nur angeboten, wenn der WMS-lite Server (wms-board-lite.mjs) läuft. Beispiele: <code>/wms?service=WMS&request=GetCapabilities</code>

Dokumente

GET /Hilfe.pdf	Wirkung: Liefert Hilfe-PDF (falls vorhanden).
----------------	---

Main-Server /api (4040)

System

GET /api/activity/status	Wirkung: Aktivitäts-/Auto-Import Status (für StatusPage Auto-Import/Idle Anzeige).
--------------------------	--

GET /api/log.csv	Wirkung: Download/Stream der Server-Log-CSV.
------------------	--

Karte - interne Generierung

DELETE /api/internal/feldkirchen-map	Query: show; hours Wirkung: Löscht Cache-Entry für die angegebene show/hours Kombination. Beispiele: /api/internal/feldkirchen-map?show=weather&hours=12
--------------------------------------	---

GET /api/internal/feldkirchen-map	Query: show=all weather (Default all); hours=<int> (Default 24) Wirkung: Erzeugt/serviert SVG-Karte (ggf. aus Cache). Beispiele: /api/internal/feldkirchen-map /api/internal/feldkirchen-map?show=weather&hours=12
-----------------------------------	--

Einsatzboard - Karten/Fahrzeuge/Gruppen

GET /api/board	Wirkung: Liefert Boardzustand (Einsätze/Karten + Spalten).
POST /api/cards	Wirkung: Neue Karte/Einsatz anlegen. Hinweis: Body enthält u.a. title, ort, typ, columnId, tolIndex, coords, alerted, isArea/areaColor ...
PATCH /api/cards/:id	Path: id Wirkung: Karte aktualisieren (Felder im Body).
POST /api/cards/:id/assign	Path: id Wirkung: Fahrzeug zu Karte zuweisen. Hinweis: Body: vehicleId
POST /api/cards/:id/move	Path: id Wirkung: Karte zwischen Spalten/Reihenfolge verschieben. Hinweis: Body: from,to,tolIndex
POST /api/cards/:id/unassign	Path: id Wirkung: Fahrzeug von Karte entfernen. Hinweis: Body: vehicleId
PATCH /api/groups/:name/availability	Path: name Wirkung: Verfügbarkeit einer Gruppe setzen.
GET /api/groups/alerted	Wirkung: Gruppen-Alarmiert-Status.
GET /api/groups/availability	Wirkung: Gruppen-Verfügbarkeiten.
GET /api/nearby	Query: cardId (pflicht); radiusKm (optional) Wirkung: Nahe Fahrzeuge/Einheiten um den Kartenstandort (Distanz-sortiert). Beispiele: /api/nearby?cardId=Einsatz-123 /api/nearby?cardId=Einsatz-123&radiusKm=12
GET /api/vehicles	Wirkung: Liefert Fahrzeuge/Einheiten Liste.
POST /api/vehicles	Wirkung: Fahrzeug anlegen. Hinweis: Body: ort,label,mannschaft,cloneOf ...
PATCH /api/vehicles/:id/availability	Path: id Wirkung: Verfügbarkeit eines Fahrzeugs setzen. Hinweis: Body: available(bool), until(ISO/ms) wenn false.
DELETE /api/vehicles/:id/position	Path: id Wirkung: Position löschen.
PATCH /api/vehicles/:id/position	Path: id Wirkung: Position setzen (Body: lat{lng oder location Felder}).

Protokoll

GET /api/protocol/	Wirkung: Liste protocol.json
POST /api/protocol/	Wirkung: Protokolleintrag anlegen. Hinweis: Body: Protokoll-Objekt (datum, zeit, infoTyp, anvon, zu, richtung, information, ergehtAn, massnahmen, ...)
GET /api/protocol/:nr	Path: nr Wirkung: Protokolleintrag (nr) lesen.
PUT /api/protocol/:nr	Path: nr Wirkung: Protokolleintrag aktualisieren.
DELETE /api/protocol/:nr/lock	Path: nr Wirkung: Bearbeitungs-Lock lösen.
POST /api/protocol/:nr/lock	Path: nr Wirkung: Bearbeitungs-Lock setzen.
POST /api/protocol/:nr/print	Path: nr Wirkung: PDF Druck für Protokoll (Server erzeugt Datei).
GET /api/protocol/:nr/print/:file	Path: nr,file Wirkung: Druckdatei abrufen.
POST /api/protocol/blank/print	Wirkung: Leeres Protokollformular drucken.
GET /api/protocol/blank/print/:file	Path: file Wirkung: Blank Druckdatei abrufen.
GET /api/protocol/csv/file	Wirkung: Download protocol.csv

Aufgaben

GET /api/aufgaben	Query: role=<ROLE> (optional; Zugriff/Override serverseitig geprüft) Header: X-Role-Id, X-User-Role, X-Auth-Role (Fallback Rolle) Wirkung: Aufgabenboard abrufen (pro Rolle). Hinweis: Auth: Session erforderlich. Beispiele: /api/aufgaben?role=S2
POST /api/aufgaben	Query: role Header: X-Role-Id, X-User-Role, X-Auth-Role Wirkung: Aufgabe anlegen (Body enthält Task-Felder).
POST /api/aufgaben/:id/edit	Path: id Query: role Header: X-Role-Id, X-User-Role, X-Auth-Role Wirkung: Aufgabe bearbeiten.
POST /api/aufgaben/:id/status	Path: id Query: role Header: X-Role-Id, X-User-Role, X-Auth-Role Wirkung: Aufgabenstatus setzen.
GET /api/aufgaben/config	Query: role Header: X-Role-Id, X-User-Role, X-Auth-Role Wirkung: Aufgaben-Config (z.B. defaultDueOffsetMinutes).
POST /api/aufgaben/ensure	Query: role Header: X-Role-Id, X-User-Role, X-Auth-Role Wirkung: Ensure/Auto-Task Erzeugung.
GET /api/aufgaben/protocols	Query: role Header: X-Role-Id, X-User-Role, X-Auth-Role Wirkung: Protokoll-Entries für Aufgabenboard (gefiltert/aufbereitet).
POST /api/aufgaben/reorder	Query: role Header: X-Role-Id, X-User-Role, X-Auth-Role Wirkung: Aufgabenreihenfolge ändern.

Mail & Scheduler

GET /api/http/schedule	Wirkung: Liste HTTP-Schedule Jobs.
------------------------	------------------------------------

<code>POST /api/http/schedule</code>	Wirkung: Neuen HTTP-Schedule Job anlegen.
<code>DELETE /api/http/schedule/:id</code>	Path: id Wirkung: HTTP-Schedule Job löschen.
<code>PUT /api/http/schedule/:id</code>	Path: id Wirkung: HTTP-Schedule Job aktualisieren.
<code>GET /api/mail/inbox</code>	Query: limit=<1..500> (Default 50) Wirkung: Liest/aggregiert Inbox-Mails bis Limit. Beispiele: <code>/api/mail/inbox?limit=100</code>
<code>GET /api/mail/inbox/status</code>	Wirkung: Status/Healthcheck für Inbox-Leser.
<code>GET /api/mail/schedule</code>	Wirkung: Liste Mail-Schedule Jobs.
<code>POST /api/mail/schedule</code>	Wirkung: Neuen Mail-Schedule Job anlegen. Hinweis: Body enthält label/to/subject/text or body + mode/time/interval + enabled etc.
<code>DELETE /api/mail/schedule/:id</code>	Path: id Wirkung: Mail-Schedule Job löschen.
<code>PUT /api/mail/schedule/:id</code>	Path: id Wirkung: Mail-Schedule Job aktualisieren.
<code>POST /api/mail/send</code>	Wirkung: Sendet E-Mail. Hinweis: Body-Parameter (JSON): to, cc, bcc, subject, text html, optional from, replyTo, headers.
<code>GET /api/mail/status</code>	Wirkung: Mail-Client Status.

Importer & Fetcher & Export

<code>GET /api/export/pdf</code>	Wirkung: Board/Listen Export als PDF (Download/stream).
<code>GET /api/ff/creds</code>	Wirkung: FF-Fetcher Credentials vorhanden?
<code>POST /api/ff/creds</code>	Wirkung: FF-Fetcher Credentials setzen.
<code>POST /api/ff/start</code>	Wirkung: FF-Fetcher starten.
<code>GET /api/ff/status</code>	Wirkung: FF-Fetcher Status.
<code>GET /api/ff/status/details</code>	Wirkung: FF-Fetcher Detailstatus.
<code>POST /api/ff/stop</code>	Wirkung: FF-Fetcher stoppen.
<code>GET /api/import/auto-config</code>	Wirkung: Auto-Import Konfiguration lesen.
<code>POST /api/import/auto-config</code>	Wirkung: Auto-Import Konfiguration speichern.
<code>POST /api/import/trigger</code>	Wirkung: Import sofort triggern.

Druck

<code>POST /api/incidents/:id/print</code>	Path: id Wirkung: PDF Druck für Einsatz/Karte.
<code>POST /api/incidents/blank/print</code>	Wirkung: Leeres Einsatzformular drucken.
<code>POST /api/print/server/print</code>	Wirkung: Server-Print Job starten (Body: fileUrl/paths + printer options)
<code>GET /api/print/server/status</code>	Wirkung: Server-Print Status

Admin & Benutzerverwaltung

<code>GET /api/admin/filtering-rules</code>	Wirkung: Filtering Rules lesen.
<code>PUT /api/admin/filtering-rules</code>	Wirkung: Filtering Rules speichern.
<code>GET /api/admin/filtering-rules/ai-analysis-config</code>	Wirkung: AI Analysis Config lesen.

`PUT /api/admin/filtering-rul` Wirkung: AI Analysis Config speichern.
`es/ai-analysis-config`

`GET /api/admin/filtering-rul` Wirkung: Learned Regeln/Statistik.
`es/learned`

`POST /api/admin/filtering-ru` Wirkung: Learned reset.
`les/reset-learned`

`GET /api/admin/filtering-rul` Wirkung: Scenario Config lesen.
`es/scenario`

`PUT /api/admin/filtering-rul` Wirkung: Scenario Config speichern.
`es/scenario`

`GET /api/admin/filtering-rul` Wirkung: Filtering Rules Status.
`es/status`

`GET /api/ui-theme` Wirkung: UI Theme lesen.

`PUT /api/ui-theme` Wirkung: UI Theme speichern (Admin).

`POST /api/user/admin/archive` Wirkung: Archiv erstellen (Admin).

`GET /api/user/admin/archive/` Wirkung: Download-Token für Archiv erstellen.
`create-download`

`GET /api/user/admin/archive/` Path: file
`download/:file` Wirkung: Archiv herunterladen.

`POST /api/user/admin/chatbot` Wirkung: Chatbot-Server starten.
`/server/start`

`POST /api/user/admin/chatbot` Wirkung: Chatbot-Server stoppen.
`/server/stop`

`POST /api/user/admin/chatbot` Wirkung: Chatbot starten.
`/start`

`GET /api/user/admin/chatbot/` Wirkung: Chatbot Status.
`status`

`POST /api/user/admin/chatbot/stop` Wirkung: Chatbot stoppen.

`POST /api/user/admin/chatbot` Wirkung: Chatbot-Worker starten.
`/worker/start`

`POST /api/user/admin/chatbot` Wirkung: Chatbot-Worker stoppen.
`/worker/stop`

`POST /api/user/admin/initialsetup` Wirkung: Initialsetup (Admin).

`GET /api/user/admin/knowledg` Wirkung: Knowledge Files Liste.
`e/files`

`DELETE /api/user/admin/knowl` Path: filename
`edge/files/:filename` Wirkung: Knowledge File löschen.

`POST /api/user/admin/knowled` Wirkung: Knowledge ingest/index.
`ge/ingest`

`POST /api/user/admin/knowled` Wirkung: Knowledge Upload (FormData: file).
`ge/upload`

`POST /api/user/admin/knowled` Wirkung: Multiple Upload (FormData: files[]).
`ge/upload-multiple`

`GET /api/user/admin/logs/dow` Wirkung: Logs herunterladen.
`nload`

`GET /api/user/admin/worker/c` Wirkung: Worker Config lesen.
`onfig`

`PATCH /api/user/admin/worker` Wirkung: Worker Config patchen.
`/config`

GET /api/user/fetcher	Wirkung: Fetcher-Creds Status.
PUT /api/user/fetcher	Wirkung: Fetcher-Creds setzen.
POST /api/user/login	Wirkung: Login (Body: username,password).
POST /api/user/logout	Wirkung: Logout.
POST /api/user/master/lock	Wirkung: Master sperren.
POST /api/user/master/setup	Wirkung: Master Setup (Body: password, adminPass, optional adminUser).
GET /api/user/master/state	Wirkung: Master vorhanden?
POST /api/user/master/unlock	Wirkung: Master entsperren.
GET /api/user/me	Wirkung: Aktueller User.
GET /api/user/online-roles	Wirkung: Online Rollen (auch vom Worker verwendbar).
GET /api/user/roles	Wirkung: Rollenliste lesen (Admin/Master abhängig).
PUT /api/user/roles	Wirkung: Rollenliste speichern.
GET /api/user/users	Wirkung: Userliste.
POST /api/user/users	Wirkung: User anlegen.
DELETE /api/user/users/:id	Path: id Wirkung: User löschen.
PATCH /api/user/users/:id	Path: id Wirkung: User patchen.

Sonstiges

POST /api/reset	Wirkung: Reset/Neuinitialisierung (Admin/operativ).
-----------------	---

Chatbot-Server /api (3001)

Simulation

GET /api/scenarios	Wirkung: Szenarienliste (Chatbot).
GET /api/scenarios/:scenarioId	Path: scenarioId Wirkung: Szenario Detail.
USE /api/sim/*	Wirkung: Proxy zum Chatbot-Server (Port 3001): alle /api/sim/* Requests werden durchgereicht.
POST /api/sim/pause	Wirkung: Simulation pausieren/fortsetzen.
GET /api/sim/scenario	Wirkung: Aktives Szenario.
POST /api/sim/start	Wirkung: Simulation starten (Body: scenarioId + options).
GET /api/sim/status	Wirkung: Simulationsstatus.
POST /api/sim/step	Wirkung: Sim: einen Schritt ausführen.
POST /api/sim/waiting-for-roles	Wirkung: Sim: waiting roles payload.

LLM

POST /api/chat	Wirkung: Chat endpoint (LLM chat).
USE /api/llm/*	Wirkung: Proxy zum Chatbot-Server (Port 3001): alle /api/llm/* Requests werden durchgereicht.
GET /api/llm/action-history	Query: limit (Default 100); offset (Default 0); category=protokoll aufgabe einsatz Wirkung: Historie ausgeführter LLM-Requests/Operationen. Beispiele: <code>/api/llm/action-history?limit=100&offset=0&category=protokoll</code>
GET /api/llm/config	Wirkung: LLM Config.
GET /api/llm/exchange/:exchangeId	Path: exchangeId Wirkung: Detail zu einem LLM Exchange.
POST /api/llm/global-model	Wirkung: Globales Modell setzen.
GET /api/llm/gpu	Wirkung: GPU Status.
GET /api/llm/model/:taskType	Path: taskType Wirkung: Modell pro TaskType.
GET /api/llm/models	Wirkung: LLM Modellliste.
GET /api/llm/ops-verworfen	Query: limit (Default 100); offset (Default 0); category=protokoll aufgabe einsatz Wirkung: Verworfene Operationen (Filter/Reparatur). Beispiele: <code>/api/llm/ops-verworfen?limit=100&offset=0</code>
GET /api/llm/profiles	Wirkung: Rate-limit Profiles.
GET /api/llm/prompt-templates	Wirkung: Prompt templates liste.
GET /api/llm/prompt-template/:name	Path: name Wirkung: Template lesen.
PUT /api/llm/prompt-template/:name	Path: name Wirkung: Template speichern.
GET /api/llm/system	Wirkung: Systemstatus.
POST /api/llm/task-config	Wirkung: Task-Config setzen.
POST /api/llm/test	Wirkung: Test-Request ans LLM.
POST /api/llm/test-model	Wirkung: Modelltest.

<code>POST /api/llm/test-with-metrics</code>	Wirkung: Test + Metrics.
<code>POST /api/llm/test-with-metrics-stream</code>	Wirkung: Streaming Test.

Situation

<code>GET /api/situation/analysis</code>	Query: role (pflicht beim Chatbot); cacheOnly=true false; forceRefresh=true false Wirkung: Proxy: Situationsanalyse abrufen oder neu erzeugen lassen (je nach Flags). Beispiele: <code>/api/situation/analysis?role=S2</code> <code>/api/situation/analysis?role=S2&cacheOnly=true</code>
<code>GET /api/situation/analysis-loop-sync</code>	Query: role (pflicht); cacheOnly=true false; forceRefresh=true false Wirkung: Analyse abrufen/neu berechnen. Beispiele: <code>/api/situation/analysis?role=S2&forceRefresh=true</code>
<code>POST /api/situation/analysis-loop-sync</code>	Wirkung: Proxy: Sync Analyse-Loop.
<code>POST /api/situation/question-loop-sync</code>	Wirkung: Analyse loop sync.
<code>POST /api/situation/question</code>	Wirkung: Proxy: Frage an KI (Body enthält role + question).
<code>POST /api/situation/question-fallback</code>	Query: role (Fallback, wenn nicht im Body) Wirkung: Frage an KI; role kommt primär aus Body.role, sonst Query.role.
<code>POST /api/situation/question-feedback</code>	Wirkung: Proxy: Feedback zu Q/A.
<code>POST /api/situation/question-feedback</code>	Wirkung: Question feedback.
<code>GET /api/situation/status</code>	Wirkung: Proxy: KI/Situation-Status.
<code>GET /api/situation/status</code>	Wirkung: Situation status (Chatbot).
<code>POST /api/situation/suggestion-feedback</code>	Wirkung: Proxy: Feedback zu Vorschlägen.
<code>POST /api/situation/suggestion-feedback</code>	Wirkung: Suggestion feedback.

Audit/Übungen

<code>DELETE /api/audit/:exerciseId</code>	Path: exerciseld Wirkung: Audit löschen.
<code>GET /api/audit/:exerciseId</code>	Path: exerciseld Wirkung: Audit Detail.
<code>POST /api/audit/end</code>	Wirkung: Audit end.
<code>POST /api/audit/events</code>	Wirkung: Audit Event anlegen.
<code>GET /api/audit/list</code>	Wirkung: Audit Liste.
<code>POST /api/audit/pause</code>	Wirkung: Audit pause.
<code>POST /api/audit/resume</code>	Wirkung: Audit resume.
<code>POST /api/audit/start</code>	Wirkung: Audit start.
<code>GET /api/audit/status</code>	Wirkung: Audit Status.

Templates

<code>GET /api/templates</code>	Wirkung: Templates Liste.
<code>POST /api/templates</code>	Wirkung: Template anlegen.

DELETE	Path: templateId
/api/templates/:templateId	Wirkung: Template löschen.
GET	Path: templateId
/api/templates/:templateId	Wirkung: Template Detail.

POST /api/templates/:templat eId/create-exercise	Path: templateId
	Wirkung: Aus Template Exercise erstellen.

Disaster-Kontext

GET	Path: disasterId
/api/disaster/:disasterId	Wirkung: Disaster Detail.
GET /api/disaster/current	Wirkung: Aktueller Disaster-Kontext.
POST /api/disaster/finalize	Wirkung: Disaster finalize.
POST /api/disaster/init	Wirkung: Disaster init.
GET /api/disaster/list	Wirkung: Disaster Liste.
POST /api/disaster/record-su ggestion	Wirkung: Suggestion record.
GET /api/disaster/summary	Query: maxLength (Default 1500) Wirkung: Zusammenfassung des Disaster-Kontexts, auf maxLength begrenzt. Beispiele: /api/disaster/summary?maxLength=1200
POST /api/disaster/update	Wirkung: Disaster update.

Feedback/Lernen

POST /api/feedback	Wirkung: Feedback speichern.
POST /api/feedback/learned-c ontext	Wirkung: Learned Context aktualisieren.
GET /api/feedback/list	Query: limit (Default 50); minRating (optional) Wirkung: Feedback Liste mit optionalem Mindest-Rating. Beispiele: /api/feedback/list?limit=100&minRating=4
POST /api/feedback/similar	Wirkung: Ähnliche Feedbacks suchen.
GET /api/feedback/stats	Wirkung: Feedback Stats.

Monitoring

GET	Wirkung: Rate-limit Statistik (Admin).
/api/admin/rate-limit-stats	
GET /api/events	Wirkung: SSE Stream für Live-Events (kein Query-Parameter).
GET /api/metrics	Wirkung: Metrics (Prometheus-like).
GET /api/metrics/stats	Wirkung: Aggregierte Metrics.

Stand: 2026-02-04 10:27 (lokal generiert aus EINFO-main (95))