

CASAVIONE

CASAVIONE
ekey bionyx Interface
für
GIRA Homeserver

Manual
Version 0.99.2

CASAVIONE ekey bionyx Interface für den GIRA Homeserver ist ein Produkt von

CASAVIONE e.K.
Obere Bergstraße 19
90607 Rückersdorf
Deutschland

Telefon: +49 911 521180
Internet: www.casavione.de
Email: info@casavione.de

Inhaber: Harald Polster
Handelsregister: Nürnberg HRA 17818
USt-IdNr.: DE309220711

Copyright 2025-2026, CASAVIONE e.K.

Inhalt

1	CASAVIONE ekey bionyx Interface.....	6
1.1	ekey bionyx Interface.....	6
1.2	Hauptmerkmale	6
1.2.1	Allgemein	6
1.2.2	Fingerscanner.....	6
1.2.3	Digitale Inputs	6
1.2.4	Sicherheit	6
1.3	Vorteile der Integration	7
1.4	Requirements.....	7
1.4.1	ekey-Controller	7
1.4.2	ekey bionyx Konto und bionyx App.....	7
1.4.3	GIRA Homeserver.....	8
1.4.4	Apps zum Übertragen der ekey-Konfiguration auf den Homeserver	8
2	Konfigurieren des ekey-Systems über die ekey bionyx-App	9
2.1	Konfigurieren	9
2.1.1	Voraussetzungen.....	9
2.1.2	Benachrichtigungs-API in der bionyx App aktivieren.....	9
2.1.3	Notification-API konfigurieren	9
2.2	Exportieren des ekey-Systems für das Logikmodul	9
2.2.1	System Mapping exportieren und mailen.....	10
2.3	HTTP(S)-Requests.....	10
2.3.1	Funktion	10
2.3.2	Anwendung	10
2.3.3	HTTP(S) Request konfigurieren	10
3	Konfiguration des Logikmoduls im Logikeditor des GIRA HS+FS Experten	15
3.1	Logikmodul importieren	15
3.2	Grafische Logikeditor	15
3.2.1	Funktionsweise	15
3.2.2	Benutzermeldungen.....	16
3.2.3	Eingänge des Logikmoduls	17
3.2.4	Ausgänge des Logikmoduls	21

4	Konfiguration des ekey-Systems an den Homeserver senden	24
4.1	Voraussetzungen	24
4.2	Exportieren der ekey-Systemkonfiguration	24
4.3	ekey-Systemkonfiguration an den Homeserver senden	25
4.3.1	Senden der Konfiguration über die App ‚ekey bionyx‘	26
4.3.2	Senden der Konfiguration über die iOS App ‚Rest‘	30
4.3.3	Senden der Konfiguration über die Android App ‚Rest Api Client‘	31
4.4	Konfiguration prüfen	31
5	Kaskadieren von Logikmodulen.....	32
5.1	Funktionalität	32
5.2	Strukturen.....	32
5.2.1	Einzelmodul	32
5.2.2	Linienstruktur	32
5.2.3	Baumstruktur.....	33
5.3	Einsatzfälle.....	34
5.4	Konfiguration	34
6	Beispiele	36
6.1	Haus mit 3 Bewohnern und einem Fingerscanner für Haustüre und Garagentor	36
6.1.1	Aufgabenstellung.....	36
6.1.2	General configuration.....	36
6.1.3	Handler configuration.....	37
7	Tipps und Tricks.....	38
7.1	Beschränkung auf 4 Finger pro Person umgehen	38
7.2	Homeserver als virtuelle Maschine	38
7.3	Relais der ekey-Controller und KNX-Aktoren in Serie schalten.....	39
7.4	Bei gelesenen Finger zusätzlich http-Request senden	39
7.5	ekey-Relais auf digitalen Input brücken	40
8	Trouble Shooting	41
8.1	Allgemein.....	41
8.2	Debug-Seite des Homeservers	41
8.3	Lokalisieren von Problemursachen	42
8.4	HTTP-Server	43
8.5	Requests	44
8.6	System Mapping.....	45

8.7	Notifications	46
8.8	HTTP-Requests	46
8.9	Verarbeitung von Request	46
8.9.1	Durch erfolgreichen Fingerscan ausgelöste Notification	46
8.9.2	Durch Fingerscan ausgelöste Notification mit eingeschränkten Berechtigungen	47
8.9.3	HTTP-Requests	47
8.9.4	Kaskadierte Logikmodule	48
9	Lizenzierung	49
9.1.1	Basislizenz	49
9.1.2	Erweiterungslizenzen	49
9.1.3	Lizenzüberprüfung	49

1 CASAVIONE ekey bionyx Interface

1.1 ekey bionyx Interface

Das CASAVIONE ekey bionyx Logikmodul ist ein Logikbaustein, der auf dem GIRA Homeserver läuft. Seine Hauptfunktion ist der Empfang und die Verarbeitung von HTTPS-Push-Nachrichten von ekey-Steuerungen. Diese Benachrichtigungen ermöglichen das Auslösen von Ereignissen, die auf bestimmten Benutzeraktionen basieren.

Das Logikmodul ist ereignisorientiert: Es empfängt Daten und löst ein Ereignis aus, das der Systemintegrator im Logikeditor des GIRA Experten definiert hat. Das Logikmodul empfängt nur Daten der ekey-Steuerungen und sendet keine Befehle oder Anfragen zurück an die ekey-Steuerungen.

1.2 Hauptmerkmale

1.2.1 Allgemein

Das Design des Logikmoduls wurde so gewählt, dass es sowohl schnell möglich ist, einfache Systeme (z.B. ein EFH mit einem Fingerscanner an der Haustüre für 3 Personen) zu konfigurieren, als auch umfangreiche Systeme mit mehreren Fingerscannern/Controllern und bis zu 100 Benutzern einzurichten.

1.2.2 Fingerscanner

- Wenn eine gültige Benachrichtigung über einen Fingerabdruck empfangen wird, gibt das Logikmodul auf dem zur Benutzer-ID und dem Fingerindex gehörenden Ausgang eine 1 aus, woraufhin beliebige Befehle im Homeserver ausgelöst werden können.
- Bei der Einrichtung können Sie einzelnen oder mehreren Fingern Ereignisse zuweisen oder mehreren Personen das gleiche Ereignis zuweisen.
- Ein Ausgang für nicht erkannte Benutzer ist verfügbar.

1.2.3 Digitale Inputs

- Wenn eine Benachrichtigung über einen digitalen Input eingeht, validiert das System die Eingabe und gibt auf dem zum digitalen Input (1 oder 2) gehörenden Ausgang eine 1 aus, woraufhin beliebige Befehle im Homeserver ausgelöst werden können. Außerdem wird die Nummer des digitalen Inputs auf einem Ausgang ausgegeben. So können bei vorhandenen ekey Erweiterungsmodulen pro Controller bis zu 18 digitale Inputs ausgewertet werden.
- Wenn eine Eingangsnummer nicht erkannt wird, wird ein allgemeines Ereignis „unbekannter Eingang“ pro Gerät ausgelöst.

1.2.4 Sicherheit

Über das Logikmodul können natürlich auch KNX-Schaltaktoren getriggert werden. So kann man auf die Idee kommen, KNX-Aktoren statt der ekey-Erweiterungsmodule einzusetzen. Dies empfiehlt sich aber aus Gründen der Sicherheit nicht, weil ein potentieller Angreifer mit Zugriff auf das KNX-System zum Beispiel die Haustüre öffnen könnte. Das heißt, aus Sicherheitsgründen

sollten unbedingt die Relais der ekey Steuerungen oder Erweiterungsmodule verwendet werden. Die Benachrichtigungen sollten nur zusätzliche Aktionen auslösen.

1.3 Vorteile der Integration

Die Integration von ekey-Geräten in den GIRA Homeserver bzw. das KNX-System bietet leistungsstarke Anpassungsmöglichkeiten für die Hausautomation und ermöglicht personalisierte Szenen und Aktivitäten. Beispiele hierfür sind:

- Routine-Automatisierung: Nutzen Sie Fingerabdrücke, um Szenen wie "Willkommen zu Hause" oder "Goodbye Mode" auszulösen, um Beleuchtung, Temperatur, Jalousien und die Alarmanlage einzustellen.
- Benutzerdefinierte Meldungen: Generieren Sie Benachrichtigungen für bestimmte Ereignisse, wie z. B. die Rückkehr der Kinder von der Schule.
- Benutzer zuhause An- und Abmelden: Melden Sie die Bewohner des Hauses beim Heimkommen im Homeserver an und bei Verlassen des Hauses ab, so dass diese Information für Automatisierungen zur Verfügung steht.
- Multi-Funktions-Zugriff: Weisen Sie verschiedenen Fingern desselben Benutzers unterschiedliche Funktionen zu. So kann beispielsweise der Zeigefinger eines Benutzers "Willkommen zu Hause" auslösen, während der Mittelfinger zusätzlich einen Alarm auslöst.
- Erhöhte Sicherheit: Implementieren Sie benutzerdefinierte Zugangskontrollen mit Benachrichtigungen beim Zutrittsversuch zu gesperrten Bereichen. Das ermöglicht eine personalisierte und sichere Hausautomatisierung.
- Das CASAVIONE Logikmodul ermöglicht Integratoren, zusammen mit den ekey Finger-scannern und dem GIRA Homeserver, maßgeschneiderte Lösungen zu erstellen, die die Sicherheit und den Komfort im Haus verbessern.

1.4 Requirements

1.4.1 ekey-Controller

ekey bietet zwei Arten von Steuereinheiten an: die dLine Control Unit und die DRM-Steuereinheit. Die dLine Control Unit wird im dLine-System verwendet und enthält 1 Relais und 1 digitalen Eingang. Die DRM-Steuereinheit wird in den Systemen xLine und sLine verwendet und enthält 2 Relais und 2 digitale Eingänge. Die DRM-Steuergeräte unterstützen Erweiterungen, die in zwei Größen erhältlich sind: mit 2 Relais und 2 Eingängen und mit 4 Relais und 4 Eingängen.

Das Logikmodul unterstützt beide Steuereinheiten, wobei der DRM-Controller auf max. 18 digitale Eingänge und Relais erweitert sein darf. Eine Anpassung der Konfiguration des Logikmoduls bei Erweiterungen ist nicht erforderlich. Sollen aber über hinzugekommene digitale Inputs Befehle ausgelöst werden, müssen diese im Homeserver entsprechend verlinkt werden.

1.4.2 ekey bionyx Konto und bionyx App

- Erforderlich ist ein ekey bionyx-Konto mit Systemadministratorrechten.
- ekey-Controller Firmware-Version 5.2 oder höher.
- Das ekey System muss für den Plus Mode konfiguriert sein.

1.4.3 GIRA Homeserver

- Für den Betrieb des CASAVIONE ekey bionyx Interface Logikmoduls ist ein Homeserver mit Version 4.11 oder neuer erforderlich. Auf älteren Versionen mit HSL 2.0 sollte das Modul laufen, die Funktionalität mit diesen wurde aber nicht getestet.
- Für die Konfiguration des Systems ist die zur Version des Homeservers passende Software ‚GIRA HS+FS Experte‘ erforderlich.

1.4.4 Apps zum Übertragen der ekey-Konfiguration auf den Homeserver

Die ekey-Konfiguration muss von der ekey bionyx App auf den Homeserver übertragen werden. In der Regel kann hierzu die ekey bionyx App verwendet werden (siehe Abschnitt 4.3.1).

Das heißt, dass keine zusätzliche App erforderlich ist.

Alternativ **kann** abhängig vom Mobilgerät zur Inbetriebnahme eine der folgenden Apps verwendet werden:

- Auf iOS-Geräten die App ‘Rest’. Die kostenfreie Version ist bereits ausreichend.
- Auf Android-Geräten die App ‘Rest API Client’.

2 Konfigurieren des ekey-Systems über die ekey bionyx-App

2.1 Konfigurieren

2.1.1 Voraussetzungen

- Erforderlich ist ein ekey bionyx-Konto mit Systemadministratorrechten.
- ekey-Controller Firmware-Version 5.2 oder höher.
- Das ekey System muss für den Plus Mode konfiguriert sein.
- Controller und Fingerprint müssen konfiguriert sein, einschließlich Funktionen und Benutzern.

Das ekey-System muss ohne Homeserver bereits funktionieren.

2.1.2 Benachrichtigungs-API in der bionyx App aktivieren

- Öffnen Sie die ekey bionyx-App und melden Sie sich an.
- Gehen Sie zu Home > Einstellungen > Smart Home-Anbindungen.
- Aktivieren Sie die Notification-API.

2.1.3 Notification-API konfigurieren

Gehen Sie zu Home > Einstellungen > Smart Home-Verbindungen > Konfiguration und setzen Sie die angezeigten Felder wie folgt:

- URL: Geben Sie die Zieladresse in folgendem Format an: „<IP des Homeservers>:<Portnummer, wie im Logikmodul>“, zum Beispiel „http://192.168.0.200:5000“.
- Timeout: Belassen Sie es bei 7s (Standard).
- Sicherheitsstufe: Wählen Sie "AllowHttp" aus der Liste.

Für die aktuelle Version des Logikmoduls, muss die Notification unverschlüsselt übertragen werden. Das ist bedingt durch die Möglichkeiten des Homeservers. Sobald neuere Softwareversionen des Homeservers den Empfang verschlüsselter Requests mit TLS unterstützen, ist vorgesehen, das Logikmodul entsprechend anzupassen.

- Authentifizierung: Klicken Sie auf ‚Sicherheitsstufe‘ und wählen Sie „Access Token“ aus der Liste. Geben Sie dann das Token ein. Dieses müssen Sie bei der Konfiguration des Logikmoduls auch eingeben.
- Ablauf in (s): Geben Sie ein, wie lange die Daten gültig sein sollen, zum Beispiel 700.000 Sekunden.
- Drücken Sie auf „Speichern“, um die Konfiguration zu speichern.

2.2 Exportieren des ekey-Systems für das Logikmodul

Die Daten der ekey-Konfiguration werden zur Konfiguration des Homeservers benötigt.

- Benutzer-Ids um benutzerspezifische Aktionen auszulösen.

- Controller-Id bzw. Fingerprint-Id wenn Aktionen vom Gerät, auf dem der Finger gescannt wurde oder auf dem ein Signal an einem Eingang eingegangen ist, abhängig sein sollen.

Diese IDs müssen im Logikmodul-Editor an die Eingänge des Logikmoduls geschrieben werden. Damit Sie die Daten bei der ersten Konfiguration des Logikmoduls im Homeserver zur Verfügung haben, empfiehlt sich wie in Abschnitt 2.2.1 beschrieben, vorzugehen.

Außerdem braucht das Logikmodul noch die vollständige Systemkonfiguration mit allen aktuellen Benutzer- und Gerätedaten. Diese können Sie nach der Grundkonfiguration des Logikmoduls, bei laufendem Homeserver als Request an das Logikmodul senden. Gehen Sie dabei wie in Abschnitt 4.1 beschrieben vor.

2.2.1 System Mapping exportieren und mailen

Führen Sie folgende Schritte aus, um die Datei zu exportieren:

- Öffnen Sie die bionyx App auf Ihrem Smart Device und melden Sie sich an.
- Navigieren Sie zu Einstellungen > Smart Home-Anbindungen > System Mapping exportieren.
- Wählen Sie ‚Mapping erstellen und teilen‘ und anschließend ‚Mail‘.

Ihr Mailprogramm wird geöffnet und die Datei system_mapping_export.json ist bereits als Anlage eingefügt.

- Geben Sie Ihre Mailadresse ein und drücken Sie ‚Senden‘ um sich die Systemkonfiguration zu senden.

2.3 HTTP(S)-Requests

2.3.1 Funktion

In der ekey bionyx App können bis zu fünf HTTP(S)-Requests konfiguriert werden. Die Requests können jeweils ein oder mehreren Benutzern und Fingern zugewiesen werden und an den Homeserver geschickt werden. Über die Requests wird einer der fünf Ausgänge Request 1-5 getriggert und/oder ein Wert als Text auf den Ausgang ‚Request Value‘ gesendet.

2.3.2 Anwendung

Requests in Verbindung mit dem Homeserver können zum Beispiel sinnvoll sein, wenn eine Gruppe von Benutzern die gleichen Aktionen im Homeserver auslösen sollen. Weisen Sie hierzu allen Benutzern der Gruppe den gleichen Request zu. So können Sie auch ohne Änderungen der Homeserverkonfiguration neue Benutzer hinzufügen.

2.3.3 HTTP(S) Request konfigurieren

- Gehen Sie zu Home > Einstellungen > Smart Home-Anbindungen > HTTP(S)-Request.
- Wählen Sie + um einen HTTP(S)-Request zu erstellen und setzen Sie die angezeigten Felder wie folgt:
- Wählen Sie ein beliebiges Symbol.
- Geben Sie einen beliebigen Namen ein, z. B. ‚Request 1‘.
- Geben Sie eine beliebige Bezeichnung ein, z. B. ‚Homeserver‘

- Wählen Sie die Methode ‚Post‘
- Geben Sie die Url in folgendem Format an: „<IP des Homeservers>:<Portnummer, wie im Logikmodul>/api/httprequest“, zum Beispiel <http://192.168.0.200:5000/api/httprequest>

- Drücken Sie ‚+‘ bei Request-Header und fügen Sie folgende Header hinzu:

- Key: Application-Type, Value: application/json
- Key: Authorization, Value: Bearer <Access-Token>. Geben Sie das Access-Token ein, das Sie auch in der ekey bionyx App für die Notifications eingegeben haben

The image shows two side-by-side screenshots of a mobile application interface. Both screens are titled 'Request-Header' and have a back arrow in the top left corner. The left screenshot, taken at 16:58, shows a form with two input fields: 'Application-Type' and 'application/json'. The right screenshot, taken at 17:01, shows a form with two input fields: 'Authorization' and 'Bearer 2211'. Both screens have a dark grey button labeled 'Bearbeiten' at the bottom.

- Drücken Sie ‚+‘ bei Request-Body und fügen Sie folgenden Body hinzu:
- Wählen Sie die Kodierung ‚application/json‘
- Um einen der Request-Ausgänge 1-5 zu triggern, fügen Sie als Inhalt folgenden json-Code ein:

```
{"type": "userrequest", "output": <Nummer des Request-Ausgangs (1-5)>}, z. B. {"type": "userrequest", "output": 1}
```
- Um einen Wert als Text auf dem Ausgang ‚Request value‘ zu senden, fügen Sie als Inhalt folgenden json-Code ein:

```
{"type": "userrequest", "value": "< Wert>"}, z. B. {"type": "userrequest", "value": "Guten Tag"}
```
- Speichern Sie die Einstellung mit ‚Speichern‘

The screenshot shows a mobile application interface with a status bar at the top displaying the time 17:02, signal strength, and battery level. Below the status bar is a back arrow icon. The main heading is "Request-Body". There are two input fields: the first contains "application/json" and the second contains a truncated JSON string {"type": "userrequest", "output": 1, "valu...". At the bottom of the screen is a dark button labeled "Speichern".

Die Request-Ausgänge 1 bis 5 und der Ausgang Request value lassen sich auch mit einem kombinierten Request ansprechen. Der Body könnte dann z. B. so aussehen:

```
{"type": "userrequest", "output": 1, "value": "Guten Tag"}
```

- Wählen Sie für die folgenden Optionen Timeout, Sicherheitsstufe und Authentifizierung die Werte, die Sie bereits bei der Konfiguration der Notification-API gewählt haben.
- Speichern Sie die Einstellung mit ‚Speichern‘

17:02

< HTTP-Request

Timeout

Timeout

6,0s

Sicherheitsstufe

AllowHttp

Authentifizierung

Access-Token

33911987

Aktiv bis 15.07.27 - 05:00:18

Ausführen mit

Controller 1

Speichern

Diese Schritte sind für jeden zu sendenden Request erforderlich.

Die Anführungszeichen müssen gerade doppelte Anführungszeichen sein (Unicode U+0022). Möglicherweise ersetzt diese Ihre Tastatur automatisch durch eines der folgenden Anführungszeichen:

- Linkes typografisches doppeltes Anführungszeichen (‘, U+201C)
- Rechtes typografisches doppeltes Anführungszeichen (’ , U+201D)
- Deutsches öffnendes doppeltes Anführungszeichen („ , U+201E)

Mit diesen Anführungszeichen, ähnlich aussehenden Zeichen oder einfachen Anführungszeichen ist der JSON-String ungültig, so dass der HTTP-Request vom Logikmodul nicht verarbeitet werden. In dem Fall werden auf der Debug-Seite die Meldungen 'Body of HTTP-request contains invalid quotation marks.' und ,Invalid data received. Request ignored.' angezeigt.

3 Konfiguration des Logikmoduls im Logikeditor des GIRA HS+FS Experten

3.1 Logikmodul importieren

Das Ihnen ausgelieferte Logikmodul wurde extra für den Homeserver mit einer bestimmten Seriennummer erstellt. Es ist somit nicht auf anderen Homeservern lauffähig.

Das Logikmodul wird als hslz-Datei ausgeliefert. Zum Importieren des Logikbausteins führen Sie folgende Schritte aus:

- Wählen Sie im Menü "Logikbausteine -> Importieren" aus.
- Wählen Sie die zu importierende hslz-Datei und drücken Sie „Öffnen“.
- Starten Sie die Experten-Software neu.

Weitere Informationen zum Importieren von Logikbausteinen finden Sie in der Dokumentation von GIRA zum Homeserver.

Wenn Sie Ihr Homeserver Projekt mit dem Logikbaustein CASAVIONE ekey bionyx Interface zum Testen in einer virtuellen Maschine laufen lassen wollen, müssen Sie das Logikmodul für die Seriennummer '000000000000' importieren. Siehe auch Abschnitt 7.2.

3.2 Grafische Logikeditor

Im grafischen Logikeditor des GIRA HS+FS Experten ist das Logikmodul im Ordner ‚CASAVIONE‘ enthalten. Öffnen oder erstellen Sie ein Logikblatt mit maximaler Größe und ziehen Sie das Modul vom Ordner auf das Logik-Blatt. Die Konfiguration der Eingänge und die Ausgänge werden in den folgenden Abschnitten beschrieben.

Um die Konfiguration übersichtlich zu gestalten, sind Ein- und Ausgänge gruppiert und mit Überschriften (z.B. – Notification --) versehen. Diese Eingänge sind ohne Funktion.

Bei den Eingängen -- Config User x -- , vereinfacht es die Konfiguration, wenn man dort den Benutzernamen bzw. eine Bezeichnung für alle Benutzer in der Liste eingibt, weil dieser ‚sprechender‘ als die UserId ist.

Auf die übliche Nummerierung E1, E2, ..., A1, ... wurde verzichtet. Wo sich Eingänge direkt auf Ausgänge auswirken, wurden diese so angeordnet, dass sie direkt gegenüber in einer Zeile positioniert sind.

3.2.1 Funktionsweise

Das Logikmodul empfängt von den ekey-Controllern gesendete Notifications bzw. Requests und wertet diese aus. Eine Notification für einen gelesenen Finger enthält u. a. folgende Informationen:

- ekey Controller
- ekey Fingerscanner
- Benutzer
- Finger

Bei der Auswertung wird geprüft, ob der Fingerscanner aus der Notification mit der Konfiguration im Baustein übereinstimmen. Wenn ja, wird die Konfiguration für den Benutzer gesucht und in dieser Gruppe die konfigurierten Finger mit dem in der Notification verglichen. Stimmen die Finger überein, wird auf dem Ausgang, der dem Eingang mit der Fingerkonfiguration gegenüberliegt eine 1 gesendet.

Die Finger der linken Hand werden mit L1, L2, L3, L4, L5, die der rechten Hand mit R1, R2, R3, R4, R5 bezeichnet. Das gilt auch, wenn in der Konfiguration abweichende Shortnames für links und rechts definiert sind (siehe weiter unten). 1 ist der Daumen, 5 der kleine Finger.

Soll eine Funktion über unterschiedliche Finger (z. B. beide Zeigefinger) getriggert werden, reicht dazu ein Ausgang. Die Finger können in einer kommaseparierten Liste definiert werden (z. B. L2,R2).

Stimmt ein Finger überein (Match), so dass eine 1 gesendet wird, wird die Behandlung der Notification nicht abgebrochen. Für die folgenden Eingänge in dieser Gruppe (für diesen Benutzer) wird weiter nach einer Übereinstimmung des Fingers gesucht. So bleiben auch Konfigurationen übersichtlich, bei denen z. B. mit Finger 2 und 3 die Präsenz im Eingang gesetzt wird, mit Finger 3 aber zusätzlich ein stiller Alarm ausgelöst werden soll.

Hierzu setzen Sie ‚Fingers for output 1‘ auf ‚R2,R3‘ und verbinden ‚Output 1‘ mit der Präsenz und ‚Fingers for output 2‘ auf ‚R3‘ und ‚Output 2‘ verbinden Sie mit dem stillen Alarm.

Mit ‚L*‘, ‚R*‘ und ‚*‘ werden alle Finger der linken Hand, alle Finger der rechten Hand bzw. alle Finger definiert. Auch Kombinationen wie L*,R2,R3 sind zulässig.

Notifications zu digitalen Inputs werden vom Logikmodul nur verarbeitet, wenn der Controller aus der Notification mit dem konfigurierten Controller im Logikmodul übereinstimmt.

3.2.2 Benutzermeldungen

Über Eingänge des Logikbausteins können Benutzermeldungen individuell definiert werden. Damit lassen sich z. B. für alle verständliche, nicht technische Meldungen in beliebigen Sprachen konfigurieren.

Für die Meldungen stehen Platzhalter zur Verfügung. Diese sind:

Platzhalter für Datums- und Zeitangaben

%d: Tag

%m: Monat

%Y: Jahr

%H: Stunde

%M: Minute

%S: Sekunde

%A: Name des Wochentags

%B: Name des Monats

%a: Abkürzungen für Wochentag (Mo, Di, ...)

%b: Abkürzungen für Monat (Jan, Feb, ...)

Die Abkürzungen stehen nur für Deutsch und Englisch zur Verfügung. Für andere Sprachen werden englische Abkürzungen verwendet. Weitere mögliche Platzhalter für Datum- und Zeitangaben finden Sie auf <https://strftime.org>.

Modulspezifische Platzhalter

%n: Name des Logikmoduls

%E: Name des ekey Devices (Fingerscanner beim Lesen eines Fingers, bzw. Controller bei digitalem Input)

%U: Benutzername

%N: Bezeichnung des Fingers (z. B. ,R2')

%I: Nummer des digitalen Inputs

%F: Funktion, die über einen digitalen Input ausgelöst wurde (z. B. ,öffnen')

%D: Detailinformation (siehe Details Mode)

3.2.3 Eingänge des Logikmoduls

Wenn der Homeserver bereits mit dem Logikmodul läuft, können Sie die Systemkonfiguration, wie in Abschnitt 4.1 beschrieben, an den Homeserver senden die Werte für User ID, Controller ID und Scanner ID für die Konfiguration aus der Debug-Seite des Homeservers kopieren. Alternativ können Sie die IDs aus der Datei system_mapping_export.json kopieren, wenn Sie sich diese, wie in Abschnitt 2.2.1 beschrieben, zugemailt haben sollten.

Die mit * markierten Eingänge werden nur beim Start des Logikmoduls gelesen. Änderungen zur Laufzeit werden ignoriert.

-- Notification --

Notification data

Eingang für Notifications bei kaskadierten Modulen (siehe Abschnitt 5).

-- Config all --

Handler-Konfiguration für alle Benutzer

All users mode

Bestimmt ob die Konfiguration in dieser Gruppe für alle Benutzer gilt, die in der aktuellen Instanz des Logikbausteins aufgeführt sind (All users mode = 1) oder für alle Benutzer im System (All users mode = 2). Bei All users mode = 0, wird kein Event in dieser Gruppe ausgelöst.

Hinweis: Sollen Notifications für gelesene Fingern für **alle ekey-Benutzer** Ausgänge in der Gruppe 'All users' triggern, **ohne dass die ekey-Benutzer explizit an den Eingängen User Id 1 bis User Id 8 gesetzt sind**, muss 'All users mode' auf **2** gesetzt sein. Bei 1 (= Default) werden die Ausgänge in diesem Fall nicht getriggert. Eine Meldung auf der Debug-Seite des Homeservers weist ggf. auf diese Situation hin.

Fingers for output 1*

Bestimmt den Finger, der beim Lesen Ausgang 1 der Gruppe All users triggert.

Fingers for output 2*

Bestimmt den Finger, der beim Lesen Ausgang 2 der Gruppe All users triggert.

Fingers for output 3*

Bestimmt den Finger, der beim Lesen Ausgang 3 der Gruppe All users triggert.

Fingers for output 4*

Bestimmt den Finger, der beim Lesen Ausgang 4 der Gruppe All users triggert.

Reserved

Reserviert für Erweiterungen.

-- Config Users 1 – bis -- Config Users 8 –

User Id*

Die UserId des Benutzers, der in dieser Gruppe behandelt werden soll. Gibt es für Benutzer identische Konfigurationen, kann hier auch eine kommaseparierte Liste von UserIds eingegeben werden.

Fingers for output 1 – 4*

Siehe oben.

Reserved

-- General configuration --

Module name

Bestimmt den Präfix für modulspezifische Einträge auf der HS-Debug-Seite. Das erleichtert die Zuordnung der Debug-Informationen zu den Logikmodulen, wenn man mehr als ein Logikmodul im Projekt verwendet. Ist der Eingang nicht explizit gesetzt, wird für das Main-Modul kein Präfix verwendet. Diese Option empfiehlt sich für das Main-Modul oder wenn es nur eine Instanz des Logikbausteins im Projekt gibt. Werden im Projekt Logikmodule kaskadiert, sollte ein Name für die kaskadierten Logikmodule definiert werden.

Bei der Definition von benutzerdefinierten Messages kann der Platzhalter %n (klein geschrieben) verwendet werden, der durch den Modulnamen ersetzt wird. Das macht normalerweise nur Sinn, wenn auch der ‚Module name‘ gesetzt wurde.

Zulässig sind folgende Zeichen: A-Z, a-z, 0-9, Ä, Ö, Ü, ä, ö, ü, ß, _ - und .

HTTP-Port*

Bestimmt den Port, auf dem der Webserver des Logikmoduls die Notifications der ekey-Controller empfängt. Der Port muss noch frei sein und mit der Konfiguration in der ekey App übereinstimmen (siehe Abschnitt 2.1.3).

Bei kaskadierten Logikmodulen, darf der Port nur im ersten Logikmodul gesetzt sein, denn nur im ersten Logikmodul darf der Webserver laufen. Für alle folgenden Logikmodule muss der HTTP-Port 0 (Default) sein.

Access-Token

Das Logikmodul verwendet das Access Token zur Authentifizierung der Herkunft von Nachrichten. Das Access-Token muss übereinstimmen mit dem Access-Token, das in der ekey bionyx App (Home >Einstellungen > Smart Home-Anbindungen > Konfiguration) festgelegt wurde. Stimmen

die Access-Tokens nicht überein, werden empfangene Notifications ignoriert. Dieser Schritt wird bei der Ersteinrichtung benötigt und wenn der Access Token in der bionyx App und im Treiber geändert werden soll.

Bei kaskadierten Logikmodule muss das Access-Token nur im ersten Logikmodul gesetzt sein.

Scanner ID

Bestimmt von welchen Scannern, in der aktuellen Instanz des Logikmoduls, Notifications zu gelesenen Fingern, behandelt werden. Eine kommaseparierte Liste mehrerer ScannerIds ist zulässig. Ist die Scanner ID ,*' (Default), behandelt der Logikbaustein die eingehenden Notifications von allen Scannern im System.

Ist die Scanner ID nicht gesetzt, verarbeitet das Logikmodul alle Notifications zu gelesenen Fingern identisch.

Enthält die Scanner ID nicht die ID in der Notification, wird die Notification in diesem Logikmodul nicht behandelt. Die Notification wird dann auch nicht an kaskadierte Module weitergeleitet. Die Scanner ID wirkt somit auch wie ein Filter für die Weiterleitung an kaskadierte Module.

Controller ID

Bestimmt von welchen Controllern, in der aktuellen Instanz des Logikmoduls, Notifications zu den digitalen Inputs behandelt werden. Eine kommaseparierte Liste mehrerer ControllerIds ist zulässig. Ist die Controller ID ,*' (Default), behandelt der Logikbaustein die Notifications aller Controller im System.

Ist die Controller ID nicht gesetzt, verarbeitet das Logikmodul alle Notifications zu den digitalen Inputs identisch.

Enthält die Controller ID nicht mit der ID in der Notification, wird die Notification in diesem Logikmodul nicht behandelt. Die Notification wird dann auch nicht an kaskadierte Module weitergeleitet. Die Controller ID wirkt somit auch wie ein Filter für die Weiterleitung an kaskadierte Module.

Handler mode

Bestimmt, in welchen Fällen die Ausgänge getriggert werden und Notifications an kaskadierte Module weitergeleitet werden. Der Wert ist bitcodiert.

Bit 0 (Wert = 1): Ausgänge werden nur getriggert, wenn der Fingerscan im ekey-System erfolgreich war.

Bit 1 (Wert = 2): Ausgänge werden nur getriggert, wenn der Fingerscan im ekey-System nicht erfolgreich war, z. B. weil der Benutzer über den Scheduler deaktiviert ist.

Bit 2 (Wert = 4): Reserviert.

Bit 3 (Wert = 8): Reserviert.

Bit 4 (Wert = 16): Erzwingt die Weiterleitung von Notifications zu gelesenen Fingern, auch wenn die Scanner ID nicht übereinstimmt.

Bit 5 (Wert = 32): Erzwingt die Weiterleitung von Notifications zu digitalen Inputs, auch wenn die Controller ID nicht übereinstimmt.

Für die ersten beiden Bits gilt somit: Beim Wert 0 werden die Ausgänge nicht getriggert, beim Werte 3 werden die Ausgänge unabhängig vom Erfolg im ekey-System getriggert.

Da sich Änderungen des Handler Modes zur Laufzeit des Homeservers sofort auswirken, lassen sich damit individuelle Zugangsberechtigungen steuern. Zum Beispiel Zugangsberechtigung für bestimmte Personen in Abhängigkeit von der Anwesenheit der Bewohner oder über die Schaltuhren des Homeservers.

Details mode

Bestimmt für den Handler mode mit gesetztem Bit 2, in welchen Fällen die Ausgänge getriggert werden.

10: Input disabled

20: Time scheduled

30: No rule found

Eine kommaseparierte Liste dieser Werte ist zulässig.

Match mode

Bestimmt wie es bei der Behandlung einer Notification und einem 'Match', d. h. wenn im Baustein der richtige Benutzer gefunden wurde (übereinstimmende UserIds) weitergeht.

0: Die Behandlung wird nach der aktuellen Gruppe abgebrochen.

1: Die Behandlung wird im aktuellen Baustein fortgesetzt, aber die Notification wird nicht an eventuell kaskadierte Module weitergegeben.

2: Die Behandlung wird im aktuellen Baustein fortgesetzt und die Notification wird an kaskadierte Module weitergeleitet.

Language/Country

Bestimmt die Sprache und Land für Meldungen im Format <Sprache>_<Land/Region>.

Zulässige Sprachen: de, en, fr, es, it, pt, nl, sv, da, fi, no, pl, cs, sk, hu, ro, bg, hr, sr, sl, et, lv, lt.

Zulässige Länder: DE, AT, CH, US, GB, FR, ES, IT, PL, NL, SV, DA, FI, NO, PL, CS, SK, HU, RO, BG, HR, SR, SL, ET, LV, LT.

Msg prefix

Bestimmt einen Prefix für alle Textausgaben. Sollen z. B. Meldungslisten erstellt werden, kann hier das Datum und die Zeit entsprechend formatiert werden (z. B. %d.%m.%Y %H:%M:%S:).

Zulässig sind nur druckbare ASCII-Zeichen (z. B. A-Z, a-z, 0-9, -, _). Werden unzulässige Zeichen verwendet, wird der eingegebene Prefix ignoriert und stattdessen der Default-Wert verwendet.

L/R finger short name

Bestimmt wie die Finger bei Meldungen ausgegeben werden. Defaultmäßig werden die Finger als L1-L5 für die Finger der linken Hand und R1-R5 für die Finger der rechten Hand ausgegeben, wobei 1 dem Daumen und 5 dem kleinen Finger zugeordnet ist. Für weitere Sprachen kann man hier andere Kürzel statt L und R festlegen. Format: <L>,<R> (z. B. „G,D“ für französisch).

Msg format fingerprint

Meldungstext bei einem erfolgreichen Fingerscan.

Msg format scheduled user

Meldungstext bei einem Fingerscan, wenn der Benutzer per Scheduler deaktiviert ist.

Msg format disabled user

Meldungstext bei einem Fingerscan, wenn diesem keine Funktion zugewiesen ist.

Msg format digital input

Meldungstext bei Notification zu digitalem Input 1 – 18.

Mit den folgenden zwei Formaten lassen sich einfach gut verständliche Meldungen erstellen, z. B. ‚Taster an Haustüre gedrückt‘.

Msg format digital input 1

Meldungstext bei Notification zu digitalem Input 1.

Msg format digital input 2

Meldungstext bei Notification zu digitalem Input 2.

Reserved**Reserved****License codes***

Enthält die Lizenzcodes für das Logikmodul. Diese sind erforderlich um die Basislizenz (bis 20 ekey-Benutzer) auf 60 bzw. 100 ekey-Benutzer zu erweitern. Die Lizenzcodes für die einzelnen Erweiterungen sind durch Komma zu trennen. Siehe auch Kapitel Lizenzierung.

Reserved**-- Logging --**

Die Eingänge der Gruppe ‘Logging’ werden nur für die Entwicklung und ggf. für den Support benötigt. D. h. die folgenden Eingänge sind normalerweise nicht zu belegen.

Logging mode**Logging level****Email address/target****Logging command****3.2.4 Ausgänge des Logikmoduls****-- Cascading --****Notification data**

Gibt Notifications und weitere Daten aus. Erforderlich zum Kaskadieren von Logikmodulen (siehe Abschnitt 5).

-- All users --

Username

Gibt den Namen des Benutzers aus. Sinnvoll z. B. zum Erstellen von Meldungen oder Begrüßungstexten.

Output 1

Sendet eine 1, wenn ein Finger gelesen wurde, der am Eingang links definiert ist.

Output 2

Sendet eine 1, wenn ein Finger gelesen wurde, der am Eingang links definiert ist.

Output 3

Sendet eine 1, wenn ein Finger gelesen wurde, der am Eingang links definiert ist.

Output 4

Sendet eine 1, wenn ein Finger gelesen wurde, der am Eingang links definiert ist.

Reserved

Reserviert

-- User 1 --

Username 1

Gibt den Namen des Benutzers aus. Sinnvoll z. B. zum Erstellen von Meldungen oder Begrüßungstexten.

Output 1

Sendet eine 1, wenn ein Finger gelesen wurde, der am Eingang links definiert ist.

Output 2

Sendet eine 1, wenn ein Finger gelesen wurde, der am Eingang links definiert ist.

Output 3

Sendet eine 1, wenn ein Finger gelesen wurde, der am Eingang links definiert ist.

Output 4

Sendet eine 1, wenn ein Finger gelesen wurde, der am Eingang links definiert ist.

Reserved

User 2 – 8 ebenso

-- Additional states --

Details

Gibt den Status Details aus, der mit der Notification vom ekey Controller gesendet wurde.

Unknown finger

Sendet eine 1, wenn ein unbekannter Finger gelesen wurde.

-- Digital Inputs --

Input 1

Sendet eine 1, wenn der digitale Input 1 getriggert wurde.

Input 2

Sendet eine 1, wenn der digitale Input 2 getriggert wurde.

Input Number

Sendet die Nummer des digitalen Inputs (1-18), wenn dieser getriggert wurde.

-- Relais --

Relais 1

Reserviert.

Relais 2

Reserviert.

Relais X

Reserviert.

-- HTTP-Requests --

Request 1-5

Sendet eine 1, wenn ein Request mit der Nummer des Ausgangs (1-5) empfangen wurde.

Request value

Sendet den Wert als Text, wenn ein Request mit einem Wert empfangen wurde.

-- Messages/Errors --

User message 1

Gibt Benutzermeldungen aus, wenn eine Notification **einen** oder mehrere Events ausgelöst hat.

User message 2

Gibt Benutzermeldungen aus, wenn eine Notification **keinen** Event ausgelöst hat.

Error state

Gibt den Fehlerstatus (0 oder 1) aus.

Last error

Gibt den letzten Fehler aus.

4 Konfiguration des ekey-Systems an den Homeserver senden

4.1 Voraussetzungen

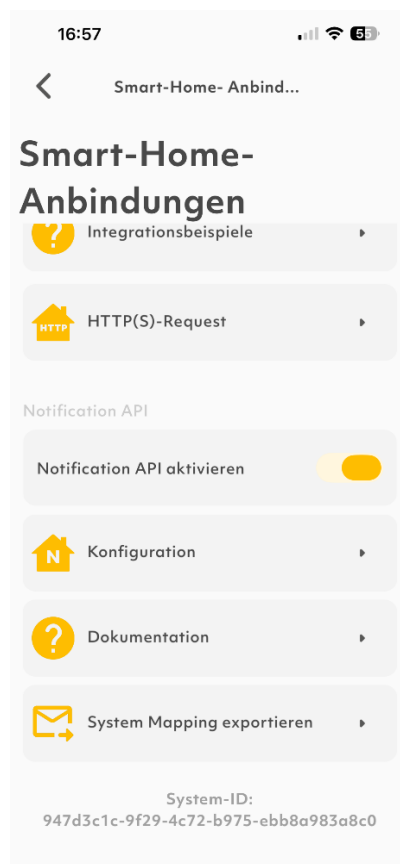
Bevor Sie die Konfiguration an den Homeserver senden, muss das Logikmodul importiert und konfiguriert (zumindest der Port und das Access-Token) sein. Der Homeserver muss gestartet sein und im Logikmodul muss der HTTP-Server laufen. Ob diese Voraussetzungen erfüllt sind, und auf welchem Port der HTTP-Server Messages empfängt können Sie auf der Debug-Seite des Homeservers prüfen. Siehe hierzu Abschnitt 8.2.

4.2 Exportieren der ekey-Systemkonfiguration

Die bionyx-App ermöglicht es Ihnen, die Systemkonfiguration mit den aktuellen Benutzer- und Gerätedaten zu exportieren. Hierzu wird eine JSON-Datei namens „system_mapping_export.json“ verwendet.

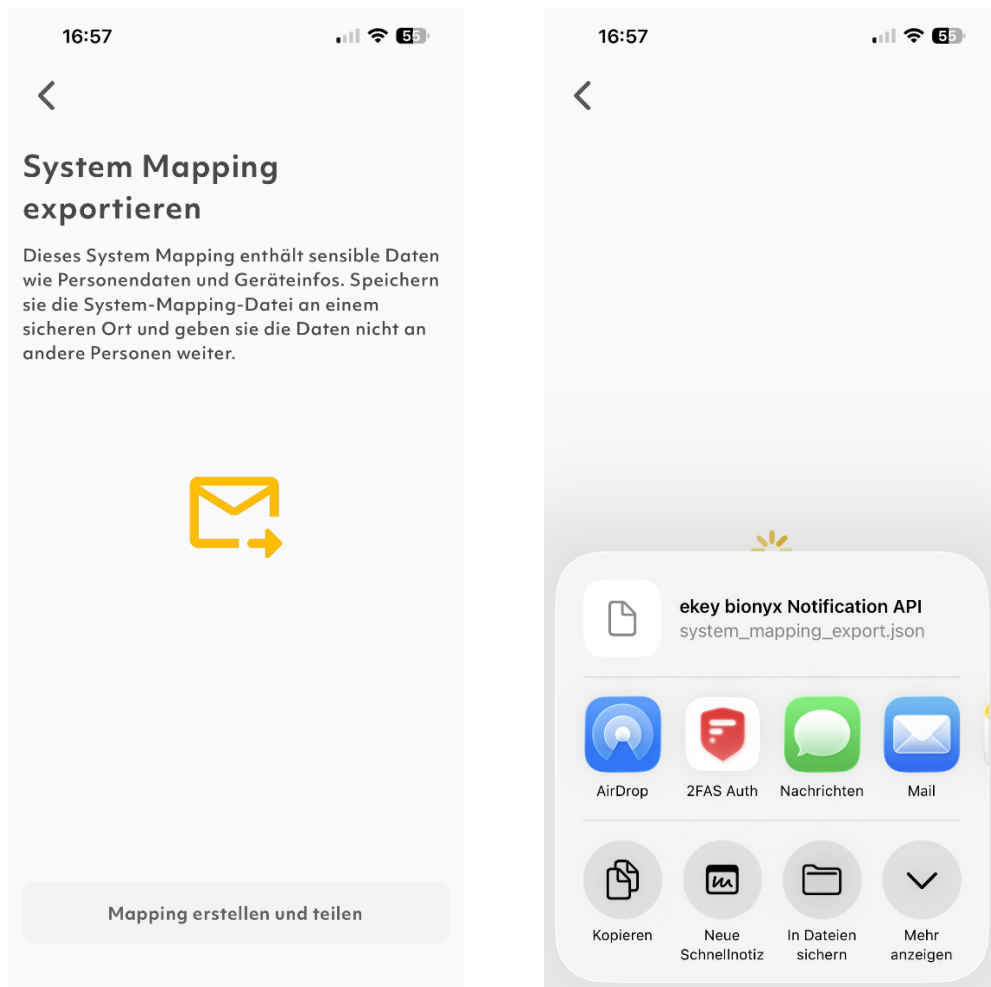
Führen Sie folgende Schritte aus, um die Datei zu exportieren:

- Öffnen Sie die bionyx App auf Ihrem Smart Device und melden Sie sich an.
- Navigieren Sie zu Einstellungen > Smart Home-Anbindungen > System Mapping exportieren.



iOS:

- Wählen Sie ‚Mapping erstellen und teilen‘ und anschließend ‚Kopieren‘ um den Inhalt der system_mapping_export.json in die Zwischenablage zu kopieren.



Android:

- Senden Sie das System-Mapping an Ihre eigene Email-Adresse.
- Öffnen Sie die Mail und speichern Sie die Datei "system_mapping_export.json".

Die Datei wird dabei unter "Eigene Dateien" -> "Downloads" gespeichert.

- Markieren Sie die Datei und nennen Sie sie in "system_mapping_export.txt" um.
- Öffnen Sie die Datei z. B. in Chrome oder einem beliebigen Texteditor.
- Wählen Sie den gesamten Text aus und kopieren Sie diesen in die Zwischenablage

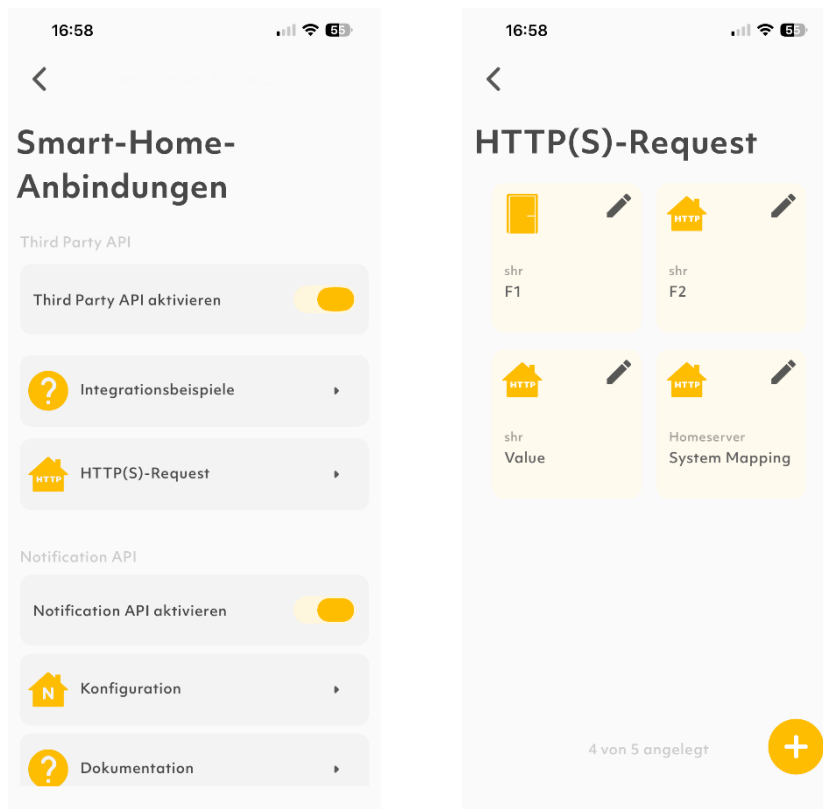
4.3 ekey-Systemkonfiguration an den Homeserver senden

Zum Senden der Konfiguration, die Sie in die Zwischenablage kopiert haben, können Sie eine der folgenden Möglichkeiten nutzen. Sollten Sie eine andere App bevorzugen, sind die Schritte anzupassen.

4.3.1 Senden der Konfiguration über die App ‚ekey bionyx‘

Die naheliegendste Möglichkeit zum Senden der Konfiguration ist, hierzu die ekey bionyx App zu verwenden und darin, wie im Folgenden beschrieben, einen HTTP-Request einzurichten und zu senden. Voraussetzung ist, dass einer der fünf HTTP-Requests zur Verfügung steht und nicht anderweitig benötigt wird. Sollte kein HTTP-Request frei sein, empfiehlt sich eine der in den folgenden Abschnitten beschriebenen Optionen einzusetzen.


Öffnen Sie die ekey bionyx App und gehen Sie zu Home > Einstellungen > Smart Home-Anbindungen > HTTP(S)-Request. Wählen Sie + um einen HTTP(S)-Request zu erstellen und setzen Sie die angezeigten Felder wie folgt:



- Wählen Sie ein beliebiges Symbol, z. B. ‚HTTP‘.
- Geben Sie einen beliebigen Namen ein, z. B. ‚System Mapping‘.
- Geben Sie einen beliebigen Ort ein, z. B. ‚Homeserver‘

16:58

< HTTP-Request








 Homeserver

 System Mappi...

Testen

Symbol

Name

System Mapping


Ort

Homeserver

- Wählen Sie die Methode ‚Post‘
- Geben Sie die Url in folgendem Format an: „<IP des Homeservers>:<Portnummer, wie im Logikmodul>/api/systemmapping“, zum Beispiel <http://192.168.0.200:5000/api/systemmapping>

17:00

< HTTP-Request








 shr

 F1

Testen

Symbol

Name

F1

Ort

shr

Methode & URL

Post

- Drücken Sie ,+' bei Request-Header und fügen Sie folgende Header hinzu:
- Key: Application-Type, Value: application/json

16:58

<

Request-Header

Application-Type

application/json

Bearbeiten

- Drücken Sie ,+' bei Request-Body und fügen Sie folgenden Body hinzu:
- Wählen Sie die Kodierung ,application/json‘
- Fügen Sie das System Mapping ein, dass Sie vorher, wie in Abschnitt 4.2, beschrieben in die Zwischenablage kopiert haben.
- Drücken Sie ,Speichern‘

16:59

<

Request-Body

application/json

```
{ "system": "Harald ÄÖÜäöüß áá!"; "us...
```

Speichern

17:01

< HTTP-Request

F1

Ort

shr

Methode & URL

Post

http://192.168.0.202:5000/api/httpreq...

Request-Header

+

{ Application-Type: application/json ✕

{ Authorization: Bearer 2211 ✕

Request-Body

{ application/json ✕

- Wählen Sie bei ‚Ausführen mit‘ den Controller, der den Request senden soll.
- Wählen Sie ‚Speichern‘, um die Konfiguration zu speichern.
- Drücken Sie oben ‚Testen‘, um den Request mit der Konfiguration des ekey-Systems an den Homeserver zu senden.

16:59

< HTTP-Request

Timeout

Timeout

6,0s

Sicherheitsstufe

AllowHttp

Authentifizierung

Access-Token

46512514

Aktiv bis 08.12.27 - 00:06:47

Ausführen mit

Controller 1

Speichern

17:00

<

HTTP-Request

Homeserver System Mappi...

Testen

Symbol

System Mapping

Ort

Homeserver

Obige Schritte müssen genauso nur bei der Ersteinrichtung ausgeführt werden. Muss das System Mapping erneut gesendet werden, z. B. nachdem ein neuer Benutzer hinzugefügt wurde, muss nur das neue System Mapping aus der Zwischenablage in das Feld 'Request Body' eingefügt, die Konfiguration gespeichert und mit 'Testen' der Request an den Homeserver gesendet werden.

4.3.2 Senden der Konfiguration über die iOS App ,Rest'

Diese Möglichkeit zum Senden der Konfiguration empfiehlt sich, wenn die Voraussetzung für das Senden über die ekey bionyx App nicht erfüllt sind (siehe Abschnitt 4.3.1.), kann aber auch sinnvoll sein, wenn Sie als Systemintegrator mit Ihrem iOS-Device mehrere Kundensysteme in Betrieb nehmen und administrieren, weil Sie sich damit das mehrfache, manuelle Erstellen der HTTP-Requests in der ekey bionyx App sparen.

Zur Vorbereitung der Übertragung müssen folgende Schritte einmalig ausgeführt werden. Die eigentliche Übertragung der Konfiguration wird im weiteren Verlauf beschrieben.

- Installieren Sie die App 'Rest' aus dem App Store
- Starten Sie die App 'Rest'
- Erstellen Sie mit '+' ein neues Projekt und geben Sie einen Namen (z. B. 'ekey') und ggf. eine kurze Beschreibung ein.
- Wählen Sie das erstellte Projekt und erstellen Sie mit '+' einen neuen Request.

Geben Sie im Tab 'Info' folgendes ein:

- Name: Ein beliebiger Name, z. B. 'Send system mapping'
- Scheme: http
- Host: Die IP-Adresse des Homeservers (z. B. 192.168.0.200)
- Port: Der Port auf dem der Homeserver auch ekey-Notifications empfängt, d. h. der Port der auch in der ekey bionyx App konfiguriert ist (z. B. 5000)
- Path: /api/systemmapping.

Geben Sie im Tab 'Headers' folgendes ein:

- Content-Type: application/json; charset=utf-8
- Authorisation: Bearer <Access-Token>. Geben Sie das Access-Token ein, das Sie auch in der ekey bionyx App für die Notifications eingegeben haben

Geben Sie im Tab 'Body' folgendes ein:

- Wählen Sie 'JSON'
- In das Textfeld können Sie später die ekey-Konfiguration system_mapping_export.json einfügen

Wenn Sie z. B. als Systemintegrator mehrere Kundensysteme mit einem Mobildevice übertragen wollen, lässt sich in der App ,Rest' ein erstellter Request durch Wischen nach links kopieren und anschließend anpassen.

Folgende Schritte sind auszuführen um die ekey System-Konfiguration an den Homeserver zu senden:

- Starten Sie die App ‚Rest‘
- Öffnen Sie das Projekt ‚ekey‘
- Wählen Sie den Request ‚Send system mapping‘
- Wählen Sie den Tab ‚Body‘.
- Fügen Sie das System Mapping ein, dass Sie vorher, wie in Abschnitt 4.2, beschrieben in die Zwischenablage kopiert haben.
- Senden Sie den Request über das Icon rechts vom Adressfeld an den Homeserver

4.3.3 Senden der Konfiguration über die Android App ‚Rest Api Client‘

Mit der Android App ‚Rest Api Client‘ oder anderen Apps ist das Vorgehen ähnlich dem in Abschnitt 4.3.2 beschriebenen, es muss aber angepasst werden.

4.4 Konfiguration prüfen

Nach dem Senden der Konfiguration können Sie auf der Debug-Seite auch prüfen, ob die Konfiguration erfolgreich übernommen wurde. Siehe hierzu Abschnitt 8.2.

**Ein eventuell ungültiges System Mapping wird auf der Debug-Seite des Homeservers angezeigt.
Ohne gültiges System Mapping kann keine Notification von ekey verarbeitet werden.**

Wichtig:

Damit das System Mapping auch nach einem Neustart des Homeservers zur Verfügung steht, wird das System Mapping im Remanentspeicher des Homeservers gespeichert. Stellen Sie sicher, dass der Homeserver nach dem Übertragen des System Mappings noch mindestens 15 Minuten läuft, weil die Remanentdaten nur alle 15 Minuten speichert werden. Sollten Sie den Homeserver vorher ausschalten, müssen Sie obige Schritte nach dem Neustart ggf. erneut ausführen.

5 Kaskadieren von Logikmodulen

5.1 Funktionalität

Im Logikmodul eingehende Notifications werden im Logikmodul verarbeitet. Abhängig von der Konfiguration des Logikbausteins werden alle Notifications oder nur die, die im aktuellen Logikmodul keine Funktion ausgeführt haben, auf den Ausgang ‚Notification data‘ ausgegeben. Diese Notifications können in kaskadierten Logikmodulen erneut ausgewertet werden und werden ggf. wieder auf deren Ausgang ‚Notification data‘ ausgegeben. Die Anzahl der kaskadierten Module ist nicht begrenzt.

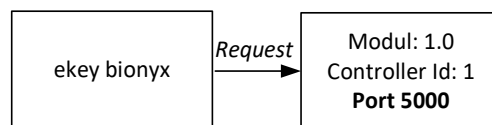
Notifications werden solange an kaskadierte Module weitergegeben, bis die Konfiguration des Moduls und die Notification bezüglich Fingerscanner/Controller, Benutzer und Finger übereinstimmen.

Ein Standardfall für den Einsatz von kaskadierten Modulen sind ein Logikmodul pro Fingerscanner/Controller.

5.2 Strukturen

5.2.1 Einzelmodul

Die einfachste Struktur ist ein einzelnes Modul ohne Kaskadierung. Diese kommt normalerweise bei genau einem Controller/Fingerscanner im System zur Anwendung.



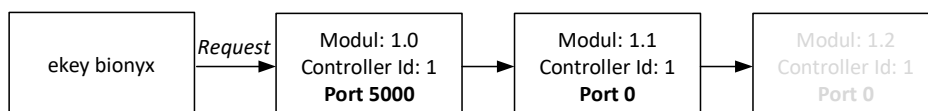
Weitere Logikmodule sind nicht erforderlich, wenn die 8 Benutzergruppen eines Logikmoduls ausreichen.

Dabei übernimmt das Modul folgende Aufgaben:

Modul 1.0: Empfängt Requests vom ekey bionyx System, wertet digitale Inputs, HTTP-Requests und Fingerscans aus.

5.2.2 Linienstruktur

Die einfachste Struktur für kaskadierte Module ist eine Linie. Diese kommt normalerweise bei genau einem Controller/Fingerscanner im System und mehr als 8 Benutzergruppen zur Anwendung.



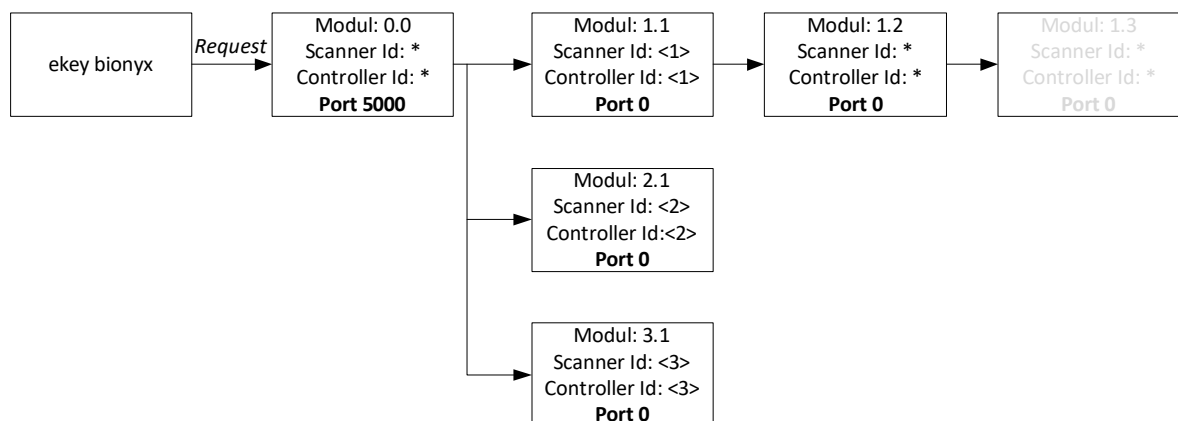
Dabei übernehmen die Module folgende Aufgaben:

Modul 1.0: Empfängt Requests vom ekey bionyx System, leitet ggf. Notifications weiter, wertet digitale Inputs, HTTP-Requests und Fingerscans aus.

Modul 1.1: Empfängt Notifications vom Modul 1.0, leitet ggf. Notifications weiter, wertet Fingerscans aus.

5.2.3 Baumstruktur

Ein ‚Verzweigen‘ von einem Logikmodul direkt auf mehrere ist ebenfalls möglich, womit sich im Projekt übersichtliche Strukturen erstellen lassen. Die folgende Abbildung zeigt eine Struktur für 3 Controller/Fingerscanner. Das Modul 0.0 empfängt die Request vom ekey bionyx System und gibt die Notifications an die Module 1.1 bis 3.1 weiter.



Im Modul 0.0 bleiben Controller ID und Scanner ID normalerweise ‚*‘, damit alle Notifications an die kaskadierten Module weitergeleitet werden. In den Modulen 1.2 bis 1.n können die IDs ebenfalls auf ‚*‘ bleiben, wenn das Modul 1.1 die Notifications für andere Module bereits ausfiltert.

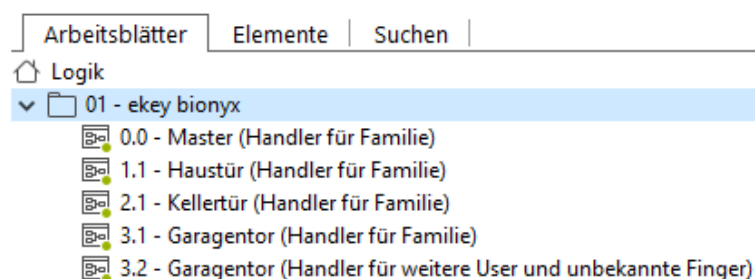
Die Module übernehmen folgende Aufgaben:

Modul 0.0: Empfängt Requests vom ekey bionyx System und leitet Notifications weiter, wertet digitale Inputs, HTTP-Requests und Fingerscans aus. Funktionen, die für alle Fingerscanner gleich sind, können in diesem Modul konfiguriert werden.

Module 1.1, 2.1 und 3.1: Empfangen Notifications vom Modul 0.0, leiten ggf. Notifications weiter, werten digitale Inputs und Fingerscans aus.

Modul 1.2: Empfängt Notifications vom Modul 1.1, leitet ggf. Notifications weiter, wertet Fingerscans aus.

Im Logikeditor sieht das, bei sinnvoller Namensgebung, übersichtlich aus.



5.3 Einsatzfälle

Für einfache Systeme mit einem Fingerscanner und wenigen Benutzern reicht oft ein Logikmodul um alle gewünschten Funktionen im Homeserver zu realisieren aus. Das kann auch für umfangreiche Systeme mit mehreren Fingerscannern und vielen Benutzern gelten, insbesondere wenn für viele Benutzer die gleichen Funktionen ausgeführt werden sollen.

Mehr als ein Logikmodul werden benötigt, wenn die 8 Benutzer(gruppen) des Logikmoduls (Eingänge Config User 1 bis Config User 8) nicht ausreichen, oder wenn für mehrere Fingerscanner unterschiedliche Funktionen ausgeführt werden sollen. Z. B. wenn mit dem gleichen Finger eines Benutzers ein ekey-Controller die Haupteingangstür öffnet, ein anderer Controller aber die Nebeneingangstür, und der Homeserver das Licht in unterschiedlichen Räumen einschalten soll.

Aber auch wenn das Kaskadieren für ein System nicht nötig ist, kann es die Übersicht im HS/FS-Logikeditor verbessern, wenn man Logikmodule kaskadiert und diese auf mehrere Logikseiten verteilt.

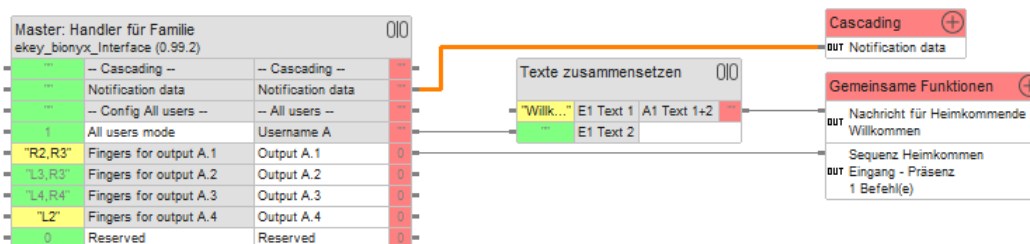
Sinnvoll ist häufig auch, ein Logikmodul für Funktionen zu nutzen, die unabhängig vom verwendeten Fingerscanner ist (z. B. um die Anwesenheit der Person im Gebäude zu setzen) und jeweils ein Modul pro Fingerscanner, über die dann die Beleuchtung in den jeweiligen Räumen geschaltet wird.

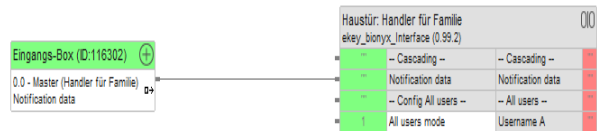
Das Kaskadieren von Logikmodulen erlaubt unter anderem:

- mehr als 8 unterschiedliche ekey-Benutzer bzw. Gruppen von Benutzern
- unterschiedliche Funktionen für jeden Fingerscanner
- verbesserte Übersichtlichkeit, z. B. indem ein Logikmodul pro Benutzer verwendet wird
- unterschiedliche Konfiguration der Logikmodule (Handler Mode, Details Mode, Match Mode)
- Verschiedene Meldungen (z. B. für Benutzermeldungen und Logs)
- Unterschiedliche Spracheinstellungen

5.4 Konfiguration

Zum Kaskadieren zweier Logikmodulen verbinden Sie im Logikeditor den Ausgang ‚Notification data‘ des ersten Moduls mit einem Konnektor. Die Bezeichnung ‚Notification data‘ bietet sich für diese Konnektoren an. Verbinden Sie den Eingang ‚Notification data‘ des zu kaskadierenden Logikmoduls mit diesem Konnektor. Stellen Sie sicher, dass Sie den Konnektor der richtigen Logikseite wählen.





Statt der Konnektoren kann man auch iKOs mit dem Typ 14 Byte-Text für die Verbindung verwenden. Dabei ist zu beachten, dass zwischen zwei Logikmodulen ein eigenes iKO verwendet werden muss, weil sonst Notifications in einer Endlosschleife zwischen den Logikmodulen gesendet werden. Diese Situation kann bei der Verwendung von Konnektoren nicht auftreten, weil der HS/FS-Experte bei der Projektprüfung diese Fehler findet und ‚Zyklus in Logik-Struktur erkannt‘ anzeigt. iKOs werden deshalb nicht zum Kaskadieren von Logikmodulen empfohlen.

Mit Ausnahme des ersten Moduls (Main), das die Requests vom ekey-System über das Netzwerk empfängt, erhalten alle kaskadierten Module die Notifications über den Eingang ‘Notification data’. Stellen Sie sicher, dass an diesen Modulen der Eingang ,Port‘ auf 0 gesetzt ist. Andernfalls würde in diesen Modulen der HTTP-Server gestartet werden und das Modul die auf diesem Port empfangenen Requests verarbeiten.

Sind an einem Logikmodul die Eingänge Controller ID und die Scanner ID gesetzt, leitet das Logikmodul Notifications nur dann an kaskadierte Module weiter, wenn die am Logikmodul gesetzte Controller ID und die Scanner ID mit den IDs der Notification übereinstimmt. Damit wird erreicht, dass im Beispiel für die Baumstruktur eine Notification vom Controller mit der ID 3 von den Modulen 0.0, 1.1, 2.1 und 3.1 verarbeitet wird, aber nicht zu 1.2, 1.3, ... gesendet wird.

Sind an einem Logikmodul die Eingänge Controller ID und die Scanner ID nicht gesetzt, leitet das Logikmodul alle Notifications an kaskadierte Module weiter. Sollen alle Notifications an kaskadierte Module weitergeleitet werden, obwohl Controller ID bzw. Scanner ID gesetzt sind, kann über die Bits 4 und 5 des Handler Modes (siehe Abschnitt 3.2.3), das Weiterleiten der Notifications erzwungen werden. Diese Option ist für bestimmte Fälle sinnvoll, sie sollte aber nicht ohne Grund verwendet werden.

6 Beispiele

6.1 Haus mit 3 Bewohnern und einem Fingerscanner für Haustüre und Garagentor

6.1.1 Aufgabenstellung

Folgende Funktionalität soll umgesetzt werden:

- Alle Personen öffnen mit Finger R2 und R3 die Haustüre (ekey). Über das Logikmodul soll die Präsenz im Eingang auf 1 gesetzt werden, woraufhin bereits vorhandene Abläufe ausgelöst werden (z. B. wenn nötig Beleuchtung einschalten).
- Alle Personen öffnen mit Finger L2 das Garagentor (Homeserver/KNX)
- Beim Heimkommen sollen die Personen auch am Haus angemeldet werden, damit der Homeserver 'weiß', wer gerade zuhause ist. Deshalb wird über Finger R2 für alle Personen auch die Anwesenheit gesetzt.
- Damit beim gemeinsamen nach Hause kommen nicht mehrere Finger gescannt werden müssen, soll über Finger R3 nicht nur die Person selbst, sondern auch deren Partner mit angemeldet werden.
- Über die Finger R4 und R5 soll auf gleiche Weise die Anwesenheit auf 0 gesetzt.

6.1.2 General configuration

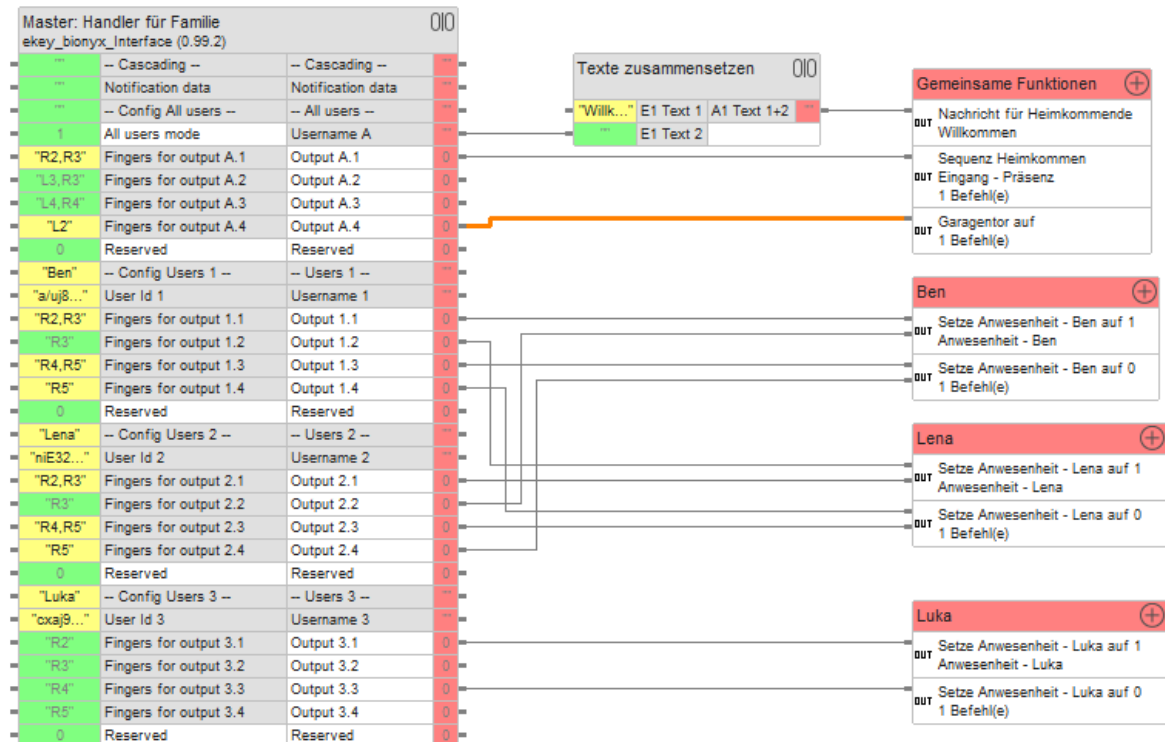
5000	HTTP-Port	Unknown finger	0	Ausgangs-Box (ID:115648) (+) OUT Unbekannter Finger 2 Befehl(e)
"2211"	Access-Token	-- Digital Inputs --		
	Reserved	Input 1	0	
	Scanner Id	Input 2	0	
	Controller Id	Input number	0	
3	Handler mode	-- Reserved --		
"10,20..."	Details mode	Reserved	0	
0	Match mode	Reserved	0	
"de_DE"	Language/country	Reserved	0	
"%d.%m..."	Msg prefix	-- HTTP-Requests --		
"L,R"	L/R finger short name	Request 1	0	
"%U ha..."	Msg format fingerprint	Request 2	0	
"Permi..."	Msg format scheduled user	Request 3	0	
"Permi..."	Msg format disabled user	Request 4	0	
"Digit..."	Msg format digital input 1	Request 5	0	
"Digit..."	Msg format digital input 2	Request value		
	Reserved	-- Messages/Errors --		
	Reserved	User message 1		
	License codes	User message 2		
	-- Logging --	Error state	0	
0	Logging mode	Last error		
1	Logging level			
"suppo..."	Email address/target			
0	Logging command			

Die Eingänge HTTP-Port und Access-Token müssen gesetzt werden. Der Handler Mode wurde auf 3 gesetzt, damit auch Notifications einen Event auslösen, die in ekey keine Aktion ausführen. Das ist

für das Setzen der Anwesenheit auf 0 nötig. Für alle weiteren Eingänge können die Defaultwerte verwendet werden.

User Message 1 und 2 werden auf **ein** iKO ausgegeben werden, da keine getrennte Ausgabe für erfolgreiche und nicht erfolgreiche Fingerscans nötig ist.

6.1.3 Handler configuration



Die gemeinsamen Funktionen werden im Abschnitt "Config All users" konfiguriert. Für die Finger R2 und R3 wird das iKO Eingang-Präsenz gesetzt, für Finger L2 das Garagentor geöffnet. Und über den Ausgang Username lässt sich noch ein Willkommenstext generieren, der z. B. auf einem Display angezeigt werden kann.

Über die folgenden Abschnitte ,Config Users 1' bis ,Config Users 3' werden die benutzerspezifischen Funktionen konfiguriert. So wird mit Finger R2 und R3 die Anwesenheit über die Ausgänge 1.1, 2.1 und 3.1 auf 1 gesetzt, und über die Ausgänge 1.3, 2.3 und 3.3 auf 0 gesetzt. Und Finger R3 setzt zusätzlich über die Ausgänge 1.2 und 2.2 die Anwesenheit des jeweiligen Partners.

Mit gleicher Configuration wird über die Finger R4 und R5 die Anwesenheit auf 0 gesetzt.

7 Tipps und Tricks

7.1 Beschränkung auf 4 Finger pro Person umgehen

Im ekey-System können für jede Person vier Finger angelernt werden. Deshalb gibt es auch im Logikmodul pro Person vier Ausgänge auf denen bei gelesenem Finger eine 1 gesendet wird. Das heißt, dass pro Person vier verschiedene Funktionen ausgelöst werden können.

Wollen Sie mehr als vier verschiedene Funktionen triggern, führen Sie folgende Schritte aus:

- Erstellen Sie in der ekey bionyx App einen weiteren Benutzer. Für diesen können Sie den gleichen Namen wie für den ursprünglichen Benutzer wählen oder einen beliebigen.
- Weisen Sie dem Benutzer die Finger zu (z. B. linker Zeigefinger – linker kleiner Finger).
- Übertragen Sie die Systemkonfiguration an den Homeserver und konfigurieren den Benutzer wie üblich im Logikmodul.

Wollen Sie weitere Finger (z. B. die Ihrer linken Hand) als Ersatz nutzen, wenn ein Finger der rechten Hand (z. B. wegen einer Verletzung) nicht erkannt wird, führen Sie folgende Schritte aus:

- Erstellen Sie in der ekey bionyx App einen weiteren Benutzer. Für diesen können Sie den gleichen Namen wie für den ursprünglichen Benutzer wählen oder einen beliebigen (z. B. ‚Max Mustermann (linke Hand)‘).
- Weisen Sie dem Benutzer die Finger zu (z. B. linker Zeigefinger – linker kleiner Finger).
- Übertragen Sie die Systemkonfiguration an den Homeserver.
- Tragen Sie am Eingang User Id (Abschnitte Config Users 1 bis Config Users 8) die User Ids beider Benutzer ein (z. B. ‚a*****1,b*****2‘).
- Tragen Sie an den Eingängen ‚Fingers for output 1‘ bis ‚Fingers for output 4‘ jeweils die Finger beider Hände ein (z. B. ‚L2,R2‘ für beide Zeigefinger).
- Selbstverständlich können die ‚Ersatzfinger‘ auch an der gleichen Hand sein, dann können Sie z. B. ‚R2,R4‘ und ‚R3,R4‘ eintragen.

Wollen Sie bis zu 10 Finger nutzen, können Sie auch mehr als 2 Benutzer pro Person erstellen.

7.2 Homeserver als virtuelle Maschine

Der Homeserver als virtuelle Maschine soll dem Inbetriebnehmer beim Erstellen und Testen von Projekten als Hilfsmittel dienen. Gira liefert mit dem HS/FS-Experten ein Image des Homeserver als virtuelle Maschine, das von VirtualBox und von VMWare als Grundlage zur Erstellung einer virtuellen Maschine verwendet werden kann. Mehr dazu finden Sie in der Dokumentation zum Homeserver im Kapitel ‚HS als virtuelle Maschine‘.

Sie können den virtuellen Homeserver verwenden um

- das Logikmodul vor dem Kauf zu Evaluieren
- nach dem Kauf des Logikmoduls bzw. der Erweiterungslizenzen für Ihren realen Homeserver, Ihre Konfiguration zu Testen

Da der virtuelle Homeserver immer die Seriennummer '000000000000' hat, benötigen Sie den Logikbaustein für diese Seriennummer. Diese Testversion mit vollem Funktionsumfang für den

Homeserver als virtuelle Maschine ist kostenfrei erhältlich. Sie wird auch beim Kauf des Logikmoduls zusammen mit dem Logikmodul für Ihren realen Homeserver ausgeliefert.

Wenn Sie Ihr Homeserver Projekt mit dem Logikbaustein CASAVIONE ekey bionyx Interface in so einer virtuellen Maschine laufen lassen wollen, müssen Sie das Logikmodul für die Seriennummer '000000000000' importieren.

Der Logikbaustein verhält sich auch ohne Lizenzcodes wie das normale Modul mit den Erweiterungen 1 und 2, also mit bis zu 100 Benutzern. Es verarbeitet die Request des ekey-Systems bis 2 Stunden nach dem Start des virtuellen Homeservers. Damit der volle Funktionsumfang wieder verfügbar ist, muss der virtuelle Homeserver neu gestartet werden. Zu beachten sind auch die generellen Einschränkungen des virtuellen Homeservers, insbesondere der fehlende Zugriff auf den KNX-Bus.

7.3 Relais der ekey-Controller und KNX-Aktoren in Serie schalten

Wie bereits in Abschnitt 1.2.4 ausgeführt, sollten aus Sicherheitsgründen unbedingt die Relais der ekey Steuerungen oder Erweiterungsmodule zum Öffnen von Türen verwendet werden, nicht aber KNX-Aktoren.

Erweiterte Möglichkeiten ergeben sich, wenn ein ekey-Relais und ein KNX-Schaltaktor in Serie geschaltet sind. Damit können Sie z. B. bestimmten Personen mit voller Zutrittsberechtigung im ekey-System, den Zutritt, abhängig von Zuständen im Homeserver/KNX, verweigern. Z. B. lässt sich so erreichen, dass Personen nur dann selbstständig Zutritt zum Objekt bekommen, wenn niemand aus der Familie zuhause oder kein Mitarbeiter im Büro ist.

7.4 Bei gelesenem Finger zusätzlich http-Request senden

Beim Lesen eines Fingers sendet der ekey-Controller normalerweise eine Notification, die auch den Benutzer und den verwendeten Finger beinhaltet. Damit lassen sich alle Szenarien konfigurieren. Bei vielen Benutzern kann die Konfiguration aber umfangreich werden, insbesondere wenn die Benutzer für gleiche Funktionen unterschiedliche Finger verwenden wollen bzw. müssen.

In vielen Fällen, lässt sich diese Situation deutlich vereinfachen, wenn man im ekey-System neben der Relaisfunktion (z. B. zum Öffnen der Tür) noch einen http-Request konfiguriert und diesen zusammen mit der Relaisfunktion den Fingern der Benutzer zuweist.

Beim Lesen des Fingers öffnet der ekey-Controller über sein Relais die Tür, sendet eine Notification mit der UserId an den Homeserver und einen http-Request, der einen der Ausgänge Request 1 bis 5 triggert. Über den Request-Ausgang können Sie z. B. die Präsenz nach der Tür setzen, wobei die Beleuchtung bei Erfordernis bereits vor dem Betreten des Bereichs eingeschaltet werden kann. Mit der Information über den Benutzer lassen sich z. B. gezielt Lichtszenen auswählen. Und das alles ohne Verwendung der verwendeten Finger im Homeserver.

Diese Möglichkeit ist durch die Anzahl der verfügbaren http-Requests im ekey-System beschränkt.

7.5 ekey-Relais auf digitalen Input brücken

Die in Abschnitt 7.4 beschriebene Situation lässt sich auch mit folgender Konfiguration effektiv lösen:

Angenommen, das Relais 1 des ekey-Controllers öffnet die Tür. Erstellen Sie eine Funktion für das Relais 2 und weisen diese zusammen mit der Türfunktion den Fingern zu. Verbinden Sie Relais 2 und den digitalen Input 2 des ekey-Controllers.

Beim Lesen des Fingers öffnet der ekey-Controller über sein Relais 1 die Tür und sendet eine Notification mit der UserId an den Homeserver. Über das Relais 2 wird der digitale Input 2 geschlossen, woraufhin der Controller eine Notification mit der Nummer des digitalen Inputs an den Homeserver sendet, woraufhin das Logikmodul den Ausgang „Input 2“ triggert. Eine mögliche Anwendung ist wieder das Beispiel in Abschnitt 7.4.

Diese Möglichkeit ist durch die Anzahl der verfügbaren Relais bzw. digitalen Inputs im ekey-System beschränkt.

8 Trouble Shooting

8.1 Allgemein

Sollte es nach der Konfiguration des Systems oder im Betrieb Probleme geben, soll Ihnen dieser Abschnitt helfen, die Ursache des Problems zu finden und es zu beseitigen. Dazu ist es wichtig, die Funktionsweise des Moduls und die Verwendung untenstehender Begriffe in diesem Dokument zu kennen. Die Verwendung stimmt auch mit der im ekey-System überein.

Notification

Eine Benachrichtigung vom ekey-Controller, ausgelöst von einem gelesenen Finger oder einem digitalen Input.

HTTP-Request

Ein in der ekey bionyx-App definierter HTTP-Request. Bis zu fünf sind in der App vom Benutzer konfigurierbar.

Request

Eine Nachricht, die sowohl eine Notification als auch ein http-Request in obigem Sinn sein kann. Ein Request wird als gültig betrachtet, wenn sein Typ dem einer ekey-Notification entspricht und das enthaltene Accesstoken dem im Logikmodul konfigurierten entspricht. Ein gültiger Request wird vom http-Server quittiert. Requests mit falschem Typ oder nicht übereinstimmendem Accesstoken werden nicht verarbeitet und nicht quittiert (beantwortet). Das kann nach Ablauf des Timeouts ggf. zu einer Fehlermeldung im sendenden System führen.

8.2 Debug-Seite des Homeservers

Auf der Debug-Seite des Homeservers werden wichtige Meldungen und Informationen (Stati) angezeigt, z. B. ob der http-Server läuft, zur Gültigkeit der Lizenz und zu eventuell aufgetretenen Fehlern. Werte die nicht global für alle Module gültig sind, werden für jede Instanz des Logikmoduls angezeigt, wobei der im Logikmodul gesetzte ‚Module name‘ in der ersten Spalte als Prefix verwendet wird. Ist der ‚Module name‘ nicht explizit gesetzt, entfällt der Prefix für das Main-Modul. Für kaskadierte Module wird als Default die Modul-ID verwendet.

CASAIONE_ekey_bionyx

Copyright	*** 2025, CASAIONE e.K. ***
M258: License expires on	01.08.2026 00:00:00
M258: Logging mode	2
M258: User group 8	-
M258: Error state	0
M258: User group 2	Veronika
M258: User group 3	cxaj9Rxd
M258: User group 1	ääÄÖÜäouß
M258: User group 6	-
M258: User group 7	-
M258: User group 4	ääÄÖÜäouß
M258: User group 5	-
M258: Last error	-
M258: HTTP-Server	Running (port 5000)
DEMO VERSION	*** NOT FOR RESALE ***
M258: Logdata sent on	-
M258: Logging level	0
M258: Log target	192.168.0.239:17005
M258: Last notification on	17.11.2025 13:24:08
M258: Last error on	-
M258: Users	Veronika (niE3231q), äääÄÖÜäouß (a/uj8Ls/)
M258: Devices	xLine Fingerprint 1 (4500380744240008), Controller REG Haustür (4500161045240004)
M258: System mapping received on	10.11.2025 17:06:24
M258: Initialized on	17.11.2025 13:21:48

[\(back\)](#)

„Status“ und „Meldungen“ auf den folgenden Seiten beziehen sich auf die Informationen der Debug-Seite des Homeservers im Abschnitt „CASAIONE_ekey_bionyx“.

8.3 Lokalisieren von Problemursachen

Sollte das System nicht wie gewünscht funktionieren, empfiehlt sich zum Lokalisieren der Ursache folgende Punkte in genau dieser Reihenfolge zu prüfen:

Datenübertragung vom ekey-Controller an das (erste) Logikmodul des Homeservers:

- Wurde der http-Server im Main-Module gestartet? Siehe Abschnitt 8.4.
- Empfängt das Modul Requests vom ekey-Controller? Siehe Abschnitt 8.5.
- Sind die Requests gültig? Siehe Abschnitt 8.5.
- Wurde das System Mapping korrekt importiert? Siehe Abschnitt 8.6.
- Werden Notifications empfangen und verarbeitet? Siehe Abschnitt 8.7.
- Werden HTTP-Requests empfangen und verarbeitet? Siehe Abschnitt 8.8.

Verarbeitung der Notifications und HTTP-Requests im ersten Logikmodul des Homeservers:

- Werden Notifications über gelesene Finger korrekt verarbeitet? Siehe Abschnitt 8.9.1.
- Werden Notifications über gelesene Finger ohne Funktion im ekey-System korrekt verarbeitet? Siehe Abschnitt 8.9.2.

- Werden http-Requests korrekt verarbeitet? Siehe Abschnitt 8.9.3

Weiterleitung zu und Verarbeitung der Notifications in kaskadierten Logikmodulen des Homeservers

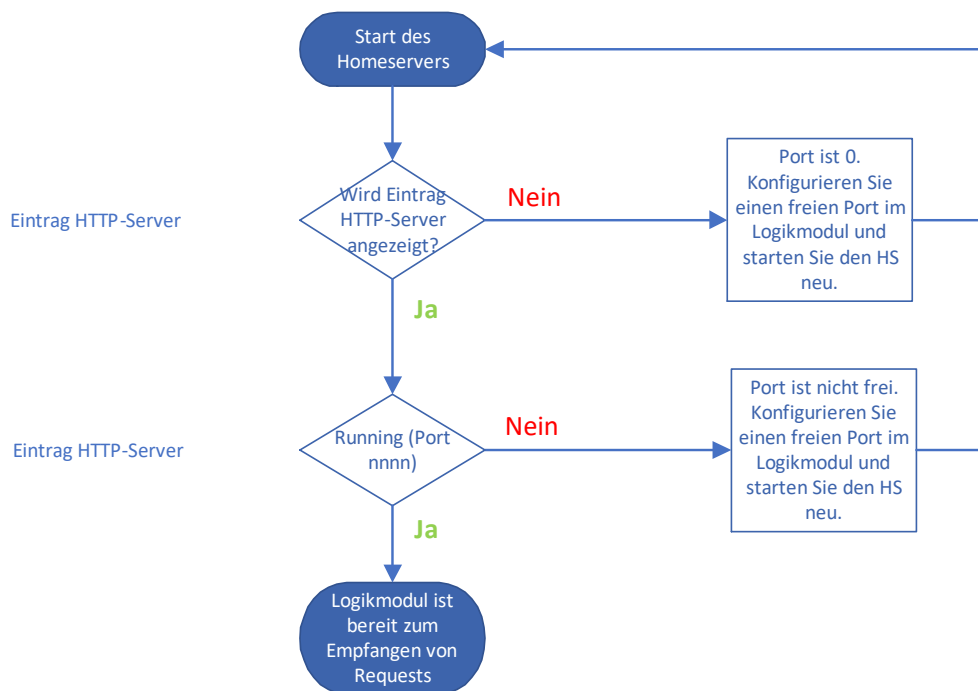
- Werden Notifications ggf. an kaskadierte Module weitergeleitet? Siehe Abschnitt 8.9.4.

Zu allen Punkten finden Sie im Folgenden detaillierte Hinweise. Zur Prüfung eines Punktes ist Voraussetzung, dass die vorherigen Punkte ok sind.

8.4 HTTP-Server

Im ersten Logikmodul (nicht aber in den kaskadierten Modulen) muss der HTTP-Server laufen, der auf dem konfigurierten Port (ungleich 0) eingehende Requests prüft und zur Verarbeitung innerhalb des Moduls und ggf. an kaskadierte Module weiterleitet.

Der HTTP-Server wird beim Neustart des Homeservers mit einer Verzögerung von 5 Sekunden gestartet. Bis dahin gibt es zum http-Server auch keine Ausgaben auf der Debug-Seite des Homeservers.



HTTP-Server: Running (Port nnnn)

Der HTTP-Server wurde gestartet und wartet auf eingehende Requests.

Status ,HTTP-Server: Start failed.‘

Meldung ,StartHttpServer: Exception when trying to start Http-Server‘

Der HTTP-Server konnte nicht gestartet werden. Grund dafür ist in der Regel, dass ein nicht verfügbarer Port konfiguriert ist.

Prüfen Sie die Ports im System und ändern Sie diese so, dass jeder Port nur einmal verwendet wird.

Es kann passieren, dass der HTTP-Server des Moduls manchmal gestartet wird, manchmal aber auch nicht. Das ist der Fall, wenn der konfigurierte Port auch für andere Dinge konfiguriert ist.

Wenn das Logikmodul den Port als erstes nutzt, startet der HTTP-Server korrekt. Ist aber ein anderer Baustein (z.B. für TCP-Empfang) früher dran, ist der Port für das ekey-Modul nicht mehr frei und der HTTP-Server kann nicht gestartet werden. Die Reihenfolge beim Start ist aber nicht unbedingt immer identisch, da beim Starten des Homeservers vielen Dinge nacheinander aber auch parallel ausgeführt werden. Durch die bewusste Verzögerung von 5 Sekunden beim Start, nutzen vermutlich alle anderen Module ihre Ports schon, so dass Konflikte schneller gefunden werden können. Am besten vergewissert man sich mit einem Port-Scanner, ob der vorgesehene Port auch tatsächlich frei ist.

Merke: Auch wenn der HTTP-Server startet, ist nicht garantiert, dass der konfigurierte Port nicht anderweitig genutzt werden soll.

8.5 Requests

Requests, die vom ekey-Controller an einen anderen als den im Logikmodul konfigurierten Port oder eine andere IP gesendet werden, können vom Logikmodul nicht erfasst werden. Für sie gibt es somit auch keine Ausgaben auf der Debug-Seite des Homeservers.

Lassen Sie den ekey-Controller einen Request an den Homeserver senden. Das können Sie auslösen durch eine der folgenden Aktionen:

- das Lesen eines (programmierten) Fingers
- das Senden eines HTTP-Requests, der für den Homeserver konfiguriert ist
- das Senden der System Mappings an den Homeserver

Wenn möglich sollten Sie die erste Aktion nutzen, weil damit das Format des Requests fix ist, und dadurch Formatfehler ausgeschlossen sind. Prüfen Sie, ob auf der Debug-Seite des Homeservers folgende Stati bzw. Meldungen angezeigt werden:

Status: Last valid request on

Gibt an, wann der letzte Request mit korrektem Content-Type und übereinstimmendem Access-Token über das Netzwerk empfangen wurde. Diesen Eintrag gibt es nur bei laufendem HTTP-Server, also wenn der Port ungleich 0 ist.

Wird dieser Eintrag nicht angezeigt, wurde vom ekey-Controller noch kein gültiger Request an den im Logikmodul konfigurierten Port gesendet.

Stati: ‚Last invalid request on‘ und ‚Invalid requests‘

Gibt an, wann der letzte ungültige Request empfangen wurde und die Anzahl dieser seit dem Start des Homeservers. Als ungültig werden Requests behandelt, wenn das Access-Token, das in der ekey bionyx App gesetzt ist, nicht mit dem im Logikmodul übereinstimmt oder der Content-Type nicht korrekt ist.

Meldung: ‚HandleHttpRequest: Request ignored due to invalid content type.‘

Ein Request mit falschem Content-Type wurde empfangen. Der Request kann nicht durch das Lesen eines Fingers oder über einen Input von einem ekey-Controller ausgelöst worden sein. Ist der Request ein vom ekey-Controller gesendeter, benutzerdefinierter HTTP-Request (z. B. mit dem System Mapping), setzen Sie dessen Content-Type auf 'application/json'. Ist das nicht der Fall, stammt der Request nicht vom ekey-System. Wenn Sie den Sender nicht lokalisieren können,

sollten Sie ggf. den Port ändern, damit im Logikmodul nur die dafür bestimmten Requests ankommen.

Meldung ,HandleHttpRequest: Request ignored due to not matching access token.'

Meldung ,HandleHttpRequest: Request ignored due to blank access token.'

Ein Request wurde empfangen, bei dem das Access-Token, das in der ekey bionyx App gesetzt ist, nicht mit dem im Logikmodul übereinstimmt oder das Access-Token leer ist.

Wenn das Modul keine Requests verarbeitet, prüfen Sie, ob das Access-Token, das in der ekey bionyx App gesetzt ist, mit dem im Logikmodul übereinstimmt.

Wenn das Modul ansonsten korrekt funktioniert, kann diese Meldung darauf hindeuten, dass auf den konfigurierten Port Requests gesendet werden, die nicht von einem ekey Controller stammen.

Wird weder ein gültiger noch ein ungültiger Request angezeigt, prüfen Sie die Einstellungen in der ekey bionyx App und stellen Sie sicher, dass die IP des Homeservers mit der konfigurierten IP in der App übereinstimmen und dass die konfigurierten Ports übereinstimmen. Stellen Sie außerdem sicher, dass keine Firewall den benutzten Port blockiert.

Wichtig: Der Fehlerstatus wird von einem erfolgreich verarbeiteten Request oder mit einem beliebigen erfolgreich ausgeführten Befehl zurückgesetzt. Sendet der ekey-Controller beim Lesen eines Fingerprints eine Notification und anschließend einen HTTP-Request, kann es zu folgender Situation kommen: Wird die Notification aufgrund einer fehlerhaften Konfiguration mit einem falschen Access-Token gesendet, wird der Fehlerstatus 1 ausgegeben. Kann der anschließend empfangene HTTP-Request korrekt verarbeitet werden, wird der Fehlerstatus wieder auf 0 gesetzt. Beachten Sie deshalb auch bei Fehlerstatus 0 alle Meldungen auf der Debug-Seite.

8.6 System Mapping

Status: System mapping received on

Gibt an, wann das aktuell verwendete System Mapping empfangen und erfolgreich verarbeitet wurde.

Status: Devices

Gibt die Gerätenamen und Geräte-Ids aus, die aus dem aktuell gültigen System Mapping gelesen wurden.

Status: Users

Gibt die Benutzernamen und Benutzer-Ids aus, die aus dem aktuell gültigen System Mapping gelesen wurden.

Mit diesen Angaben können Sie prüfen, ob die Systemkonfiguration korrekt aus ekey übernommen wurde und den richtigen Stand hat.

Meldung ,ReadSystemMapping: System mapping is empty'

Ein gültiges System Mapping steht nicht zur Verfügung, es wurde noch nicht aus der ekey bionyx App übertragen, oder der Homeserver hat es nach der Übertragung nicht remanent gespeichert.

Übertragen Sie das System Mapping aus der ekey bionyx App an den Homeserver (siehe Abschnitt 4.1).

Meldung ‚ReadSystemMapping: System mapping is corrupted‘

Das Format des empfangenen System Mapping ist nicht ok. Übertragen Sie das System Mapping aus der ekey bionyx App **unverändert und vollständig** an den Homeserver (siehe Abschnitt 4.1).

8.7 Notifications

Lassen Sie den ekey-Controller eine Notification an den Homeserver senden. Das können Sie auslösen durch eine der folgenden Aktionen:

- das Lesen eines (programmierten) Fingers
- das Schließen des Kontakts eines digitalen Eingangs des ekey-Controllers, z. B. über einen daran angeschlossenen Taster

Last notification on

Gibt an, wann zuletzt eine Notification (Finger gelesen oder digitaler Input) empfangen und verarbeitet wurde.

Wird hier kein Wert angezeigt, wurde nach dem Start des Homeservers noch keine Notification an das Logikmodul geschickt. Prüfen Sie die Einstellungen in der ekey bionyx App.

8.8 HTTP-Requests

Lassen Sie den ekey-Controller einen http-Request an den Homeserver senden. Das können Sie in der ekey bionyx App auslösen durch eine der folgenden Aktionen:

- auf der Hauptseite des Benutzers, durch Gedrückt halten eines http-Buttons
- unter Einstellungen -> Smarthome Anbindungen -> HTTP(S)-Request -> <Request wählen> -> Testen

Die erste Variante ist schneller zugänglich, bei der zweiten sehen Sie dafür gleich alle Einstellungen und können diese überprüfen.

Last HTTP-request on

Gibt an, wann der zuletzt einer der fünf benutzerdefinierbaren HTTP-Requests empfangen und verarbeitet wurde.

8.9 Verarbeitung von Request

Gültige Requests, also Notifications und http-Requests werden vom http-Server zur Verarbeitung im Logikmodul und Notifications mit gelesenen Fingern ggf. an kaskadierte Module weitergeleitet.

Bei Problemen mit Notifications sollten Sie zuerst die Funktionsweise im Main-Module prüfen, und anschließend ggf. kaskadierte Module.

8.9.1 Durch erfolgreichen Fingerscan ausgelöste Notification

Lassen Sie den ekey-Controller eine Notification an den Homeserver senden, indem Sie mit dem Fingerprinter einen programmierten Finger lesen. Diesem muss eine Funktion zugewiesen sein und der Benutzer darf nicht über die Zeitschaltuhr eingeschränkt sein.

Prüfen Sie folgende Punkte:

1. Kommt die Notification am Logikmodul an? (siehe Abschnitt 8.7)
2. Sendet das Logikmodul am Ausgang für den gelesenen Finger des Benutzers eine 1?
3. Gibt das Logikmodul eine Nachricht an den Ausgängen User Message 1 oder 2 aus?

Wenn die Notification am Logikmodul ankommt, aber das Logikmodul keine 1 sendet und keine Meldung ausgibt, prüfen Sie folgende Einstellungen an den Eingängen des Logikmoduls:

- Handler Mode: Darf nicht 0 sein.
- User Id des Benutzers: Muss mit der gesendeten Id übereinstimmen.
- Konfigurierte Finger: Muss mit dem gelesenen Finger übereinstimmen.
- War das Lesen des Fingers im ekey-System erfolgreich? Siehe Zutrittsprotokolle in der ekey bionyx App.

Wird eine Meldung auf dem Ausgang User Message 1 ausgegeben, wurde auch immer eine 1 auf einen Ausgang gesendet.

8.9.2 Durch Fingerscan ausgelöste Notification mit eingeschränkten Berechtigungen

Soll auf einem Ausgang eine 1 gesendet werden, auch wenn der Benutzer z. B. aufgrund eines zugewiesenen Zeitplans keine Funktion im ekey-System ausführt? Um das zu prüfen, lassen Sie den ekey-Controller eine Notification an den Homeserver senden, indem Sie mit dem Fingerprinter einen programmierten Finger lesen. Diesem muss eine Funktion zugewiesen sein, aber der Benutzer soll über den Zeitplan eingeschränkt sein.

Prüfen Sie folgende Punkte:

1. Kommt die Notification am Logikmodul an? (siehe Abschnitt 8.7)
2. Sendet das Logikmodul am Ausgang für den gelesenen Finger des Benutzers eine 1?
3. Gibt das Logikmodul eine Nachricht an den Ausgängen User Message 1 oder 2 aus?

Wenn die Notification am Logikmodul ankommt, aber das Logikmodul keine 1 sendet und keine Meldung ausgibt, prüfen Sie folgende Einstellungen an den Eingängen des Logikmoduls:

- Handler Mode. Muss 2 oder 3 sein.
- Details Mode: Muss für Zeitschaltuhr 20 sein, für deaktivierten Input 10 und bei nicht zugewiesener Funktion 30. Ggf. 10, 20, 30.
- User Id des Benutzers
- Konfigurierte Finger

8.9.3 HTTP-Requests

Führen HTTP-Requests nicht zum gewünschten Ergebnis, prüfen Sie folgende Punkte:

- Kommt der HTTP-Request am Logikmodul an? (siehe Abschnitt 8.8)
- Stimmt das Format des HTTP-Requests? (siehe Abschnitt 2.3.3)

8.9.4 Kaskadierte Logikmodule

Führen Notifications in kaskadierten Modulen nicht zum gewünschten Ergebnis, prüfen Sie folgende Punkte:

- Kommt die Notification am Main-Logikmodul an? (siehe Abschnitt 8.7)
- Ist das kaskadierte Logikmodul über ein **eigenes** iKO mit dem vorangehenden Logikmodul verbunden?
- Wurde für die User Id des Benutzers bereits in vorangehenden Logikmodulen eine Aktion ausgeführt? Wenn ja, muss dort der Match Mode auf 2 gesetzt sein.
- Gibt das vorangehende Logikmodul die Notification aus? Dies können Sie mit den üblichen Möglichkeiten des Homeservers prüfen.

9 Lizenzierung

9.1.1 Basislizenz

Das Ihnen ausgelieferte Logikmodul wurde extra für den Homeserver mit einer bestimmten Seriennummer erstellt. Es ist somit nicht auf anderen Homeservern lauffähig. Damit dieses Modul mit der Basislizenz läuft ist kein Lizenzschlüssel nötig.

- Die Basislizenz unterstützt bis zu 20 aktive Benutzer.

9.1.2 Erweiterungslizenzen

Die Erweiterungslizenzen ermöglichen es Ihnen, die Anzahl der aktiven Benutzer über das Limit der Basislizenz hinaus zu erhöhen.

- Erweiterungslizenz 1: Erweitert das System auf maximal 60 aktive Benutzer.
- Erweiterungslizenz 2: Erweitert das System auf maximal 100 aktive Benutzer. Erfordert eine gültige Lizenz für die Erweiterung 1. Ist diese nicht vorhanden, wird die Erweiterungslizenz 2 ignoriert und der Baustein läuft mit der Limitierung der Basislizenz.

9.1.3 Lizenzüberprüfung

Die Lizenzen werden nur beim Start des Homeservers geprüft. Das bedeutet, dass das Senden von Lizenzcodes an den Eingang ‚License code‘ des Logikbausteins zur Laufzeit des Homeservers den Lizenzstatus nicht ändert.

Geben Sie nach dem Kauf des/der Lizenzschlüssels den/die erhaltenen Schlüssel direkt am Eingang ‚License code‘ des Logikbausteins ein und übertragen die Änderungen an den Homeserver.

Nach dem Start des Homeservers, wird der aktive Typ der Lizenz auf der Debug-Seite des Homeservers angezeigt. Außerdem wird die Anzahl der Benutzer im System und die max. zulässige Anzahl angezeigt.

Die Anzahl der Benutzer wird beim Start des Homeservers aus der `system_mapping_export.json` gelesen. Außerdem wird die Anzahl der Benutzer zur Laufzeit mit jeder empfangenen Notification aktualisiert.