Přístupový systém

Merit Access

Dveřní jednotka

DoorUnit v.3.5

Firmware verze 032

25.2.2026

CZ

SDK v1.0

Důležité upozornění

Z bezpečnostních důvodů by měl být systém projektován a instalován kvalifikovaným technikem, znalým elektrotechnických předpisů a norem.

Před vlastní montáží si prosím pozorně přečtěte tento manuál, abyste předešli chybám, které mohou vést k nefunkčnosti nebo dokonce k poškození zařízení.

Všechna elektrická propojení provádějte pouze při odpojeném napájecím zdroji.

Jakékoliv zásahy do konstrukce a opravy jsou zakázané a zaniká tak právo na reklamaci výrobku.

Cílem společnosti MERIT GROUP a.s. je neustále zlepšovat kvalitu produktů, což může vést k rozdílným technickým specifikacím a firmwaru. Aktuální informace o výrobku a změnách jsou dostupné na webových stránkách.

http://meritaccess.cz

Prohlášení os shodě najdete na www.meritaccess.cz/ce

Obsah

Overview 4
Komunikační schémata 4
MQTT 4
Komunikace přes SOAP 6
Komunikace pomocí API 7
Příloha 1 8
Příloha 2 13

Overview

Toto SDK popisuje komunikační rozhraní jednotky, které je možno využít pro komunikaci a ovládání jednotky. Není určeno koncovým uživatelům a může být změněno bez upozornění v závislosti na vývoji nových verzí hw, firmwaru či navazujících sw.

Pro vývoj integračních projektů je vhodné kontaktovat výrobce a dohodnout vhodnou spolupráci.

Výrobce neručí za zachování popisovaných postupů v budoucích verzích.

Komunikační schémata

Jednotka Merit DoorUnit podporuje 3. možné způsoby komunikace.

MQTT – této komunikační model je implementován z důvodu částečné kompatibility z dřívějšími HW verzemi jednotky. V defaultu je zakázán, je nutno jej v nastavení jednotky povolit a nastavit patřičný mqtt topic. Default je /ma/. Pro komunikaci používá jednotka topic /ma/common tam posílá odpovědi na příkazy. Příkazy je potřeba zasílat na topic /ma/<název jednotky> např. MDUXXXXXXXXXXX význam příkazů zachycuje následující popis

- C00 nastaveni casu jednotky např. “C00 2021/12/01 13:58:00”
- C01 reset jednotky
- C02 otevreni dveri např. “C02100001”
  - 1000 delka pulsu v ms
  - 01,02 normalni puls
  - 03,04 rele vypnuto
  - 05,06 rele sepnuto
  - 07,08 prohozeni stavu rele
- C03 vrati aktuálni cas
- napr. “X12|MDU6827199FA906|1638367457” - cas je 1638367457 = Wednesday, December 1, 2021 2:04:17 PM
- https://www.epochconverter.com/
- C10 vlozeni karty do EEPROM C10XXXXYYYYZZZZQQQQRRTT
  - XXXX - desitkove poradi 1-1000,
  - YYYY - celkem z 1-1000
  - ZZZZ - hexa fc,
  - QQQQ - hexa id,
  - RR ctecka 01,02
  - TT casovy plan 01-62
  - napr. C1000010001691F3F710200
  - nahraje jednu kartu z jedné,
  - pro reader 2
  - cas. plan 00
  - karta 26911 0016241 (691F 3F71)hex
- C11 zapis do EEPROM C11XXXXXYYzzzzz…..
  - XXXXX adresa dekadicky (od 0)
  - YY pocet 1-32 deka
  - zzzz…. data po bytech v HEXA
  - např. C11000000101 - nastaveni offline rezimu
- C12 cteni z EEPROM C12XXXXXYY
- XXXXX adresa dekadicky
- YY pocet
- napr. C120000001 - zjisteni rezimu - bzte na adrese 00000
- C13 vypis stavu DU
  - vysledek “X13|MDU6827199FA906|Up:1” v minutách

- C14 verze firmware jednotky
    - “X14|MDU6827199FA906|Ver:6.6.6”
- C15 stav buff1/buff2/log
    - “X15|MDU6827199FA906|1000/1/0”
- C16 stejn0 jako C10, navic obsahuje 1 byte, ktery udava CRC8 z fc a id - 4 byte
    - pokud po obdrzeni cisla karty jednotka spocita CRC8 a pokud neni stejne jako obdrzené vyhlasi CRC chybu a kartu nenehraje, do budoucna to bude asi jediná podporovaná možnost jak nahrát kartu do jednotky
    - např. “C1600010001691F3F71020037” - 37 je CRC8
- C17,C18 vrati stav vstupu na magnet
- C19,C20 vrati stav rele pro zamek - reflektuje jen zmeny pomoci C02 - prikazy 03-08 NE PROSTY PULS

Modře označené oblasti nejsou ve vezi 3. implementovány, platí pouze pro verzi 2. Přestože je MQTT komunikace jednoduchá na integraci, je to něco co považujeme za součást minulosti a je zde velký předpoklad, že v budoucnu nebude implementována ani v omezeném rozsahu.

Komunikace přes SOAP

Jednotka v režimu cloud kontaktuje nastavený server z následujícím SOAP rozhraním. Adresu serveru je potřeba nastavit v nastavení jednotky. Ve výchozím nastavení je to cloudová služba Merit Access na adrese https://ws.meritaccess.cz/service?WSDL, pokud implementuje programátor níže uvedené funkce a nastaví adresu svého serveru, bude jednotka komunikovat s jeho WS. Popis funkci je možno stáhnout na uvedené adrese, nebo je v příloze opis.

Stručný popis jednotlivých funkcí

public string SQLReady()

- Ta je jednotkou volána pro ověření funkce a spojení na WS

public string GetAllCardsForTerminal(string TerminalName)

- Je volána jednotkou pro naplněni tabulky cards tzn. načtení karet s přístupem do jednotky (je volána v nepravidelných intervalech – cca jednou denně (ten random čas je tam úmyslně aby se například při nějakém globálním restartu všechny jednotky nezačali plnit a nepřetížili server) nebo při zjištěni neaktuálnosti viz. níže

public string GetTimeZonesForTerminal(string TerminalName)

- stejné jako GetAllCardsForTerminal s tím, že slouží k naplnění tabulky časových plánů
- Obě metody fungují transakčně, tzn. vytvoří se pomocná tabulka a v rámci transakce se vymění za původní, aby se eliminovala situace neúplných nebo nekonzistentních dat – bylo důležité zejména u verze 2 (tam nebyl podklad na mysql...)

public string OpenDoorOnline(string TerminalName, string card, int reader, string kdy)

- Online validace přístupu viz. níže popis funkce

public string InsertToAccess(string TerminalName, string card, int reader, string kdy, [Optional] int? status)

- Vloží do cloudu informaci o přístupu včetně stavu, jak to dopadlo viz. popis funkce

Jak jednotka komunikuje

- Po startu jednotka – pokud je v režimu cloud – bliká či svítí zeleně - pomocí funkce **SQLReady()** zjistí, zda je funkční spojení na ws a pokud není, v pravidelných intervalech test opakuje
- Pokusí se udělat synchronizaci svých dat pomocí volání **GetAllCardsForTerminal** a **GetAllCardsForTerminal** – toto nemusí být nutně hned po startu, na to má nějaký mechanizmus, kdy to dělá kvůli zátěži serveru
- Jednotka čeká na událost (přiložení karty)
- Po přiložení karty jednotka vyhodnotí na základě lokální databáze, zda má karta přístup
    - Pokud ANO – provede otevření a zároveň zavolá funkci **OpenDoorOnline,** aby se ujistila, že i cloud má stejné informace, pokud dojde ke zjištění, že nikoliv vyvolá aktualizaci dat pomocí **GetAllCardsForTerminal**
    - Pokud NE - zavolá funkci **OpenDoorOnline,** aby se ujistila, že i cloud má stejné informace, pokud zjistí, že cloud má informaci, že přístup je povolen, stihne ještě otevřít a opět vyvolá aktualizaci dat pomocí **GetAllCardsForTerminal**
- Následuje pokus o zápis informací o průchodu pomocí **InsertToAccess**

Pokud je jednotka mimo dosah cloudu, označuje si v lokální DB které přístupy nepředala na cloud (využívá k tomu specifická čísla ve sloupci status) a po připojení provede pomocí **InsertToAccess** jejich přelití na cloud.

Komunikace pomocí API

Pro popis a otestovaní si stáhněte soubor apispec.json a importujte do

https://editor.swagger.io/ Mějte na paměti, že uvedené API se primárně používá pro integraci s driverem C4 a funkce jsou tudíž přizpůsobeny tomuto driveru a do budoucna se jistě budou měnit v závislosti na požadavcích driveru. Výpis apispec.json v příloze 2.

Příloha 1

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:
tns="http://tempuri.org/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmln
s:http="http://schemas.microsoft.com/ws/06/2004/policy/http" xmlns:msc="htt
p://schemas.microsoft.com/ws/2005/12/wsdl/contract" xmlns:wsp="http://schem
as.xmlsoap.org/ws/2004/09/policy" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd" xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata" xmlns:w
sdl="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://tempuri.org/
" name="ISoapService">
<wsdl:types>
<xsd:schema elementFormDefault="qualified" targetNamespace="http://tempuri.
org/">
<xsd:import namespace="http://schemas.microsoft.com/2003/10/Serialization/A
rrays"/>
<xsd:import namespace="http://schemas.datacontract.org/2004/07/System"/>
<xsd:element name="SQLReady">
<xsd:complexType/>
</xsd:element>
<xsd:element name="SQLReadyResponse">
<xsd:complexType>
<xsd:sequence>
<xsd:element minOccurs="0" maxOccurs="1" name="SQLReadyResult" type="xsd:st
ring"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="GetAllCardsForTerminal">
<xsd:complexType>
<xsd:sequence>
<xsd:element minOccurs="0" maxOccurs="1" name="TerminalName" type="xsd:stri
ng"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="GetAllCardsForTerminalResponse">
<xsd:complexType>
<xsd:sequence>
<xsd:element minOccurs="0" maxOccurs="1" name="GetAllCardsForTerminalResult
" type="xsd:string"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="InsertToAccess">
<xsd:complexType>
<xsd:sequence>
<xsd:element minOccurs="0" maxOccurs="1" name="TerminalName" type="xsd:stri
ng"/>
<xsd:element minOccurs="0" maxOccurs="1" name="card" type="xsd:string"/>
<xsd:element minOccurs="1" maxOccurs="1" name="reader" type="xsd:int"/>
<xsd:element minOccurs="0" maxOccurs="1" name="kdy" type="xsd:string"/>
<xsd:element nillable="true" minOccurs="1" maxOccurs="1" name="status" type
="xsd:int"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="InsertToAccessResponse">
```

```
<xsd:complexType>
<xsd:sequence>
<xsd:element minOccurs="0" maxOccurs="1" name="InsertToAccessResult" type="
xsd:string"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="OpenDoorOnline">
<xsd:complexType>
<xsd:sequence>
<xsd:element minOccurs="0" maxOccurs="1" name="TerminalName" type="xsd:stri
ng"/>
<xsd:element minOccurs="0" maxOccurs="1" name="card" type="xsd:string"/>
<xsd:element minOccurs="1" maxOccurs="1" name="reader" type="xsd:int"/>
<xsd:element minOccurs="0" maxOccurs="1" name="kdy" type="xsd:string"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="OpenDoorOnlineResponse">
<xsd:complexType>
<xsd:sequence>
<xsd:element minOccurs="0" maxOccurs="1" name="OpenDoorOnlineResult" type="
xsd:string"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="GetTimeZonesForTerminal">
<xsd:complexType>
<xsd:sequence>
<xsd:element minOccurs="0" maxOccurs="1" name="TerminalName" type="xsd:stri
ng"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="GetTimeZonesForTerminalResponse">
<xsd:complexType>
<xsd:sequence>
<xsd:element minOccurs="0" maxOccurs="1" name="GetTimeZonesForTerminalResul
t" type="xsd:string"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="GetTerminalSettings">
<xsd:complexType>
<xsd:sequence>
<xsd:element minOccurs="0" maxOccurs="1" name="TerminalName" type="xsd:stri
ng"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="GetTerminalSettingsResponse">
<xsd:complexType>
<xsd:sequence>
<xsd:element minOccurs="0" maxOccurs="1" name="GetTerminalSettingsResult" t
ype="xsd:string"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>
</wsdl:types>
<wsdl:message name="ISoapService_SQLReady_InputMessage">
```

```
<wsdl:part name="parameters" element="tns:SQLReady"/>
</wsdl:message>
<wsdl:message name="ISoapService_SQLReady_OutputMessage">
<wsdl:part name="parameters" element="tns:SQLReadyResponse"/>
</wsdl:message>
<wsdl:message name="ISoapService_GetAllCardsForTerminal_InputMessage">
<wsdl:part name="parameters" element="tns:GetAllCardsForTerminal"/>
</wsdl:message>
<wsdl:message name="ISoapService_GetAllCardsForTerminal_OutputMessage">
<wsdl:part name="parameters" element="tns:GetAllCardsForTerminalResponse"/>
</wsdl:message>
<wsdl:message name="ISoapService_InsertToAccess_InputMessage">
<wsdl:part name="parameters" element="tns:InsertToAccess"/>
</wsdl:message>
<wsdl:message name="ISoapService_InsertToAccess_OutputMessage">
<wsdl:part name="parameters" element="tns:InsertToAccessResponse"/>
</wsdl:message>
<wsdl:message name="ISoapService_OpenDoorOnline_InputMessage">
<wsdl:part name="parameters" element="tns:OpenDoorOnline"/>
</wsdl:message>
<wsdl:message name="ISoapService_OpenDoorOnline_OutputMessage">
<wsdl:part name="parameters" element="tns:OpenDoorOnlineResponse"/>
</wsdl:message>
<wsdl:message name="ISoapService_GetTimeZonesForTerminal_InputMessage">
<wsdl:part name="parameters" element="tns:GetTimeZonesForTerminal"/>
</wsdl:message>
<wsdl:message name="ISoapService_GetTimeZonesForTerminal_OutputMessage">
<wsdl:part name="parameters" element="tns:GetTimeZonesForTerminalResponse"/
>
</wsdl:message>
<wsdl:message name="ISoapService_GetTerminalSettings_InputMessage">
<wsdl:part name="parameters" element="tns:GetTerminalSettings"/>
</wsdl:message>
<wsdl:message name="ISoapService_GetTerminalSettings_OutputMessage">
<wsdl:part name="parameters" element="tns:GetTerminalSettingsResponse"/>
</wsdl:message>
<wsdl:portType name="ISoapService">
<wsdl:operation name="SQLReady">
<wsdl:input message="tns:ISoapService_SQLReady_InputMessage"/>
<wsdl:output message="tns:ISoapService_SQLReady_OutputMessage"/>
</wsdl:operation>
<wsdl:operation name="GetAllCardsForTerminal">
<wsdl:input message="tns:ISoapService_GetAllCardsForTerminal_InputMessage"/
>
<wsdl:output message="tns:ISoapService_GetAllCardsForTerminal_OutputMessage
"/>
</wsdl:operation>
<wsdl:operation name="InsertToAccess">
<wsdl:input message="tns:ISoapService_InsertToAccess_InputMessage"/>
<wsdl:output message="tns:ISoapService_InsertToAccess_OutputMessage"/>
</wsdl:operation>
<wsdl:operation name="OpenDoorOnline">
<wsdl:input message="tns:ISoapService_OpenDoorOnline_InputMessage"/>
<wsdl:output message="tns:ISoapService_OpenDoorOnline_OutputMessage"/>
</wsdl:operation>
<wsdl:operation name="GetTimeZonesForTerminal">
<wsdl:input message="tns:ISoapService_GetTimeZonesForTerminal_InputMessage"
/>
<wsdl:output message="tns:ISoapService_GetTimeZonesForTerminal_OutputMessag
e"/>
</wsdl:operation>
```

```
<wsdl:operation name="GetTerminalSettings">
<wsdl:input message="tns:ISoapService_GetTerminalSettings_InputMessage"/>
<wsdl:output message="tns:ISoapService_GetTerminalSettings_OutputMessage"/>
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="BasicHttpBinding_ISoapService_soap" type="tns:ISoapServ
ice">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
<wsdl:operation name="SQLReady">
<soap:operation soapAction="http://tempuri.org/ISoapService/SQLReady" style
="document"/>
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetAllCardsForTerminal">
<soap:operation soapAction="http://tempuri.org/ISoapService/GetAllCardsForT
erminal" style="document"/>
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="InsertToAccess">
<soap:operation soapAction="http://tempuri.org/ISoapService/InsertToAccess"
 style="document"/>
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="OpenDoorOnline">
<soap:operation soapAction="http://tempuri.org/ISoapService/OpenDoorOnline"
 style="document"/>
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetTimeZonesForTerminal">
<soap:operation soapAction="http://tempuri.org/ISoapService/GetTimeZonesFor
Terminal" style="document"/>
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetTerminalSettings">
<soap:operation soapAction="http://tempuri.org/ISoapService/GetTerminalSett
ings" style="document"/>
```

```
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="ISoapService">
<wsdl:port name="BasicHttpBinding_ISoapService_soap" binding="tns:BasicHttp
Binding_ISoapService_soap">
<soap:address location="https://ws.meritaccess.cz/service"/>
</wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

Příloha 2


```
{
 "basePath": "/",
 "definitions": {},
 "info": {
  "description": "Swagger docs for Door Unit Flask API",
  "title": "Door Unit API",
  "version": "1.0.0"
 },
 "paths": {
  "/api/": {
   "get": {
    "description": "Returns all registered API routes.\n",
    "produces": [
     "application/json"
    ],
    "responses": {
     "200": {
      "description": "Success",
      "examples": {
       "application/json": {
        "routes": [
         "/api/",
         "/api/keepalive",
         "/api/restart",
         "/api/config",
         "/api/events",
         "/api/login",
         "/api/door/setoutput",
         "/api/door/setupdoor",
         "/api/door/getdoorstatus",
```

```
      "/api/credentials/card",
      "/api/credentials/setcalendar",
      "/api/credentials/removecalendar",
      "/docs",
      "/apidocs/index.html",
      "/docs/apispec.json"
    ]
   }
  },
  "schema": {
   "properties": {
    "routes": {
     "items": {
      "type": "string"
     },
     "type": "array"
    }
   },
   "required": [
    "routes"
   ],
   "type": "object"
  }
 },
 "401": {
  "description": "Unauthorized (invalid or missing HMAC headers)",
  "schema": {
   "properties": {
    "AcknowledgeType": {
     "example": 1,
     "type": "integer"
```

```
          },
          "Error": {
            "example": "Unauthorized",
            "type": "string"
          }
        },
        "type": "object"
      }
    }
  },
  "security": [
    {
      "HMACAuth": []
    }
  ],
  "summary": "List available API endpoints",
  "tags": [
    "root"
  ]
 }
},
"/api/config": {
 "get": {
  "description": "Returns MeritAccess configuration including AccessPanel settings and Doors configuration.\n",
  "produces": [
    "application/json"
  ],
  "responses": {
    "200": {
      "description": "Success",
```

```
"examples": {
  "application/json": {
    "AcknowledgeType": 0,
    "Config": {
      "AccessPanel": {
        "AppUpdate": "meritaccess",
        "CurrentIP": "192.168.1.16",
        "Dg": "192.168.10.1",
        "Dhcp": true,
        "Dns1": "8.8.8.8",
        "EnableIvar": false,
        "ID": "MDUD83ADDB52EB8",
        "Ip": "192.168.10.201",
        "IvarServer": "http://10.10.10.1",
        "IvarTerminalName1": "term1",
        "IvarTerminalName2": "term2",
        "LastStart": "2026-02-07T11:24:42.222054",
        "LogLevel": "debug",
        "Mask": "255.255.255.0",
        "Mode": 1,
        "MqttEnabled": false,
        "MqttServer": "192.168.10.1",
        "MqttTopic": "/ma/",
        "SyslogServer": "",
        "UpdateMode": 0,
        "Version": "v1.1.1",
        "Ws": "https://ws.meritaccess.cz/service?WSDL"
      },
      "Doors": {
        "1": {
          "HasMonitor": false,
```

```
        "MaxOpenTime": 30000,
        "MonitorDefault": false,
        "Output": "relay",
        "Protocol": "wiegand",
        "PulseTimeMS": 3000,
        "SystemTimePlan": 0
       },
       "2": {
        "HasMonitor": false,
        "MaxOpenTime": 30000,
        "MonitorDefault": false,
        "Output": "relay",
        "Protocol": "wiegand",
        "PulseTimeMS": 3000,
        "SystemTimePlan": 0
       }
     }
    },
    "Error": "None"
  }
 },
 "schema": {
  "properties": {
   "AcknowledgeType": {
    "example": 0,
    "type": "integer"
   },
   "Config": {
    "properties": {
     "AccessPanel": {
      "additionalProperties": true,
```

```
        "description": "Access panel configuration",
        "type": "object"
       },
       "Doors": {
        "additionalProperties": {
         "additionalProperties": true,
         "type": "object"
        },
        "description": "Map of doorId -> door configuration",
        "type": "object"
       }
      },
      "required": [
       "AccessPanel",
       "Doors"
      ],
      "type": "object"
     },
     "Error": {
      "example": "None",
      "type": "string"
     }
    },
    "required": [
     "AcknowledgeType",
     "Config",
     "Error"
    ],
    "type": "object"
   }
  },
```

```
"401": {
 "description": "Unauthorized (invalid or missing HMAC headers)",
 "schema": {
  "properties": {
   "AcknowledgeType": {
    "example": 1,
    "type": "integer"
   },
   "Error": {
    "example": "Unauthorized",
    "type": "string"
   }
  },
  "required": [
   "AcknowledgeType",
   "Error"
  ],
  "type": "object"
 }
},
"500": {
 "description": "Failed",
 "schema": {
  "properties": {
   "AcknowledgeType": {
    "example": 1,
    "type": "integer"
   },
   "Error": {
    "example": "Failed",
    "type": "string"
```

```
        }
       },
       "required": [
        "AcknowledgeType",
        "Error"
       ],
       "type": "object"
      }
     }
    },
    "security": [
     {
      "HMACAuth": []
     }
    ],
    "summary": "Get configuration",
    "tags": [
     "root"
    ]
   },
   "put": {
    "consumes": [
     "application/json"
    ],
    "description": "Updates the device configuration.\n\n\u26a0\ufe0f Applying configuration triggers restart.\n",
    "parameters": [
     {
      "in": "body",
      "name": "body",
      "required": true,
```

```
"schema": {
 "example": {
  "ConfigDU": {
   "AppUpdate": "meritaccess",
   "Dg": "192.168.10.1",
   "Dhcp": true,
   "Dns1": "8.8.8.8",
   "EnableIvar": false,
   "Ip": "192.168.10.201",
   "IvarServer": "http://10.10.10.1",
   "IvarTerminalName1": "term1",
   "IvarTerminalName2": "term2",
   "LogLevel": "debug",
   "Mask": "255.255.255.0",
   "Mode": 1,
   "MqttEnabled": false,
   "MqttServer": "192.168.10.1",
   "MqttTopic": "/ma/",
   "SyslogServer": "",
   "UpdateMode": 0,
   "Ws": "https://ws.meritaccess.cz/service?WSDL"
  },
  "Readers": {
   "1": {
    "HasMonitor": false,
    "MaxOpenTime": 30000,
    "MonitorDefault": false,
    "Output": "relay",
    "Protocol": "wiegand",
    "PulseTimeMS": 3000,
    "SystemTimePlan": 0
```

```
    },
    "2": {
      "HasMonitor": false,
      "MaxOpenTime": 30000,
      "MonitorDefault": false,
      "Output": "relay",
      "Protocol": "wiegand",
      "PulseTimeMS": 3000,
      "SystemTimePlan": 0
    }
  }
},
"properties": {
  "ConfigDU": {
    "additionalProperties": true,
    "description": "Device configuration",
    "type": "object"
  },
  "Readers": {
    "additionalProperties": {
      "additionalProperties": true,
      "type": "object"
    },
    "description": "Map of readerId \u2192 reader configuration",
    "type": "object"
  }
},
"required": [
  "ConfigDU",
  "Readers"
],
```

```
      "type": "object"
    }
  }
],
"produces": [
  "application/json"
],
"responses": {
  "200": {
    "description": "Configuration applied",
    "schema": {
      "properties": {
        "AcknowledgeType": {
          "example": 0,
          "type": "integer"
        },
        "Error": {
          "example": "None",
          "type": "string"
        }
      },
      "required": [
        "AcknowledgeType",
        "Error"
      ],
      "type": "object"
    }
  },
  "401": {
    "description": "Unauthorized (invalid or missing HMAC headers)",
    "schema": {
```

```
    "properties": {
     "AcknowledgeType": {
      "example": 1,
      "type": "integer"
     },
     "Error": {
      "example": "Unauthorized",
      "type": "string"
     }
    },
    "required": [
     "AcknowledgeType",
     "Error"
    ],
    "type": "object"
   }
  },
  "500": {
   "description": "Failed",
   "schema": {
    "properties": {
     "AcknowledgeType": {
      "example": 1,
      "type": "integer"
     },
     "Error": {
      "example": "Failed",
      "type": "string"
     }
    },
    "required": [
```

```
          "AcknowledgeType",
          "Error"
         ],
         "type": "object"
        }
       }
      },
      "security": [
       {
        "HMACAuth": []
       }
      ],
      "summary": "Set configuration",
      "tags": [
       "root"
      ]
     }
    },
    "/api/credentials/card": {
     "delete": {
      "consumes": [
       "application/json"
      ],
      "description": "NOT IMPLEMENTED - Removes an existing card credential from the system.\n",
      "parameters": [
       {
        "in": "body",
        "name": "body",
        "required": true,
        "schema": {
```

```
    "example": {
     "CardNumber": "1234567890",
     "ReaderId": 1
    },
    "type": "object"
   }
  }
 ],
 "produces": [
  "application/json"
 ],
 "responses": {
  "200": {
   "description": "Card deleted successfully",
   "schema": {
    "properties": {
     "AcknowledgeType": {
      "example": 0,
      "type": "integer"
     },
     "Error": {
      "example": "None",
      "type": "string"
     }
    },
    "type": "object"
   }
  },
  "401": {
   "description": "Unauthorized (invalid or missing HMAC headers)",
   "schema": {
```

```
    "properties": {
     "AcknowledgeType": {
      "example": 1,
      "type": "integer"
     },
     "Error": {
      "example": "Unauthorized",
      "type": "string"
     }
    },
    "type": "object"
   }
  },
  "500": {
   "description": "Failed",
   "schema": {
    "properties": {
     "AcknowledgeType": {
      "example": 1,
      "type": "integer"
     },
     "Error": {
      "example": "Failed",
      "type": "string"
     }
    },
    "type": "object"
   }
  }
 },
 "security": [
```

```
    {
      "HMACAuth": []
    }
  ],
  "summary": "NOT IMPLEMENTED - Delete card",
  "tags": [
    "not-implemented"
  ]
 },
 "patch": {
  "consumes": [
    "application/json"
  ],
  "description": "NOT IMPLEMENTED - Updates an existing card credential.\n",
  "parameters": [
    {
      "in": "body",
      "name": "body",
      "required": true,
      "schema": {
        "example": {
          "AccessLevel": 2,
          "CardNumber": "1234567890",
          "Enabled": true,
          "ReaderId": 1
        },
        "type": "object"
      }
    }
  ],
  "produces": [
```

```
  "application/json"
 ],
 "responses": {
  "200": {
   "description": "Card updated successfully",
   "schema": {
    "properties": {
     "AcknowledgeType": {
      "example": 0,
      "type": "integer"
     },
     "Error": {
      "example": "None",
      "type": "string"
     }
    },
    "type": "object"
   }
  },
  "401": {
   "description": "Unauthorized (invalid or missing HMAC headers)",
   "schema": {
    "properties": {
     "AcknowledgeType": {
      "example": 1,
      "type": "integer"
     },
     "Error": {
      "example": "Unauthorized",
      "type": "string"
     }
```

```
      },
      "type": "object"
    }
   },
   "500": {
    "description": "Failed",
    "schema": {
     "properties": {
      "AcknowledgeType": {
       "example": 1,
       "type": "integer"
      },
      "Error": {
       "example": "Failed",
       "type": "string"
      }
     },
     "type": "object"
    }
   }
  },
  "security": [
   {
    "HMACAuth": []
   }
  ],
  "summary": "NOT IMPLEMENTED - Update card",
  "tags": [
   "not-implemented"
  ]
 },
```

```
"post": {
 "consumes": [
  "application/json"
 ],
 "description": "NOT IMPLEMENTED - Adds a new card credential to the system.\n",
 "parameters": [
  {
   "in": "body",
   "name": "body",
   "required": true,
   "schema": {
    "example": {
     "AccessLevel": 1,
     "CardNumber": "1234567890",
     "FacilityCode": 10,
     "ReaderId": 1
    },
    "type": "object"
   }
  }
 ],
 "produces": [
  "application/json"
 ],
 "responses": {
  "200": {
   "description": "Card added successfully",
   "schema": {
    "properties": {
     "AcknowledgeType": {
      "example": 0,
```

```
      "type": "integer"
     },
     "Error": {
      "example": "None",
      "type": "string"
     }
    },
    "type": "object"
   }
  },
  "401": {
   "description": "Unauthorized (invalid or missing HMAC headers)",
   "schema": {
    "properties": {
     "AcknowledgeType": {
      "example": 1,
      "type": "integer"
     },
     "Error": {
      "example": "Unauthorized",
      "type": "string"
     }
    },
    "type": "object"
   }
  },
  "500": {
   "description": "Failed",
   "schema": {
    "properties": {
     "AcknowledgeType": {
```

```
          "example": 1,
          "type": "integer"
        },
        "Error": {
          "example": "Failed",
          "type": "string"
        }
      },
      "type": "object"
    }
   }
  },
  "security": [
   {
    "HMACAuth": []
   }
  ],
  "summary": "NOT IMPLEMENTED - Add card",
  "tags": [
   "not-implemented"
  ]
 }
},
"/api/credentials/removecalendar": {
 "post": {
  "consumes": [
   "application/json"
  ],
  "description": "NOT IMPLEMENTED - Removes or unassigns a calendar.\n",
  "parameters": [
   {
```

```
    "in": "body",
    "name": "body",
    "required": true,
    "schema": {
     "example": {
      "CalendarId": 1
     },
     "type": "object"
    }
   }
  ],
  "produces": [
   "application/json"
  ],
  "responses": {
   "200": {
    "description": "Calendar removed successfully",
    "schema": {
     "properties": {
      "AcknowledgeType": {
       "example": 0,
       "type": "integer"
      },
      "Error": {
       "example": "None",
       "type": "string"
      }
     },
     "type": "object"
    }
   },
```

```
"401": {
 "description": "Unauthorized (invalid or missing HMAC headers)",
 "schema": {
  "properties": {
   "AcknowledgeType": {
    "example": 1,
    "type": "integer"
   },
   "Error": {
    "example": "Unauthorized",
    "type": "string"
   }
  },
  "type": "object"
 }
},
"500": {
 "description": "Failed",
 "schema": {
  "properties": {
   "AcknowledgeType": {
    "example": 1,
    "type": "integer"
   },
   "Error": {
    "example": "Failed",
    "type": "string"
   }
  },
  "type": "object"
 }
```

```
    }
   },
   "security": [
    {
     "HMACAuth": []
    }
   ],
   "summary": "NOT IMPLEMENTED - Remove calendar",
   "tags": [
    "not-implemented"
   ]
  }
 },
 "/api/credentials/setcalendar": {
  "post": {
   "consumes": [
    "application/json"
   ],
   "description": "NOT IMPLEMENTED - Adds or assigns a calendar to credentials.\n",
   "parameters": [
    {
     "in": "body",
     "name": "body",
     "required": true,
     "schema": {
      "example": {
       "CalendarId": 1,
       "Days": [
        "Mon",
        "Tue",
        "Wed",
```

```
        "Thu",
        "Fri"
      ],
      "From": "08:00",
      "Name": "Working hours",
      "To": "18:00"
    },
    "type": "object"
  }
 }
],
"produces": [
 "application/json"
],
"responses": {
 "200": {
  "description": "Calendar added successfully",
  "schema": {
   "properties": {
    "AcknowledgeType": {
     "example": 0,
     "type": "integer"
    },
    "Error": {
     "example": "None",
     "type": "string"
    }
   },
   "type": "object"
  }
 },
```

```
"401": {
 "description": "Unauthorized (invalid or missing HMAC headers)",
 "schema": {
  "properties": {
   "AcknowledgeType": {
    "example": 1,
    "type": "integer"
   },
   "Error": {
    "example": "Unauthorized",
    "type": "string"
   }
  },
  "type": "object"
 }
},
"500": {
 "description": "Failed",
 "schema": {
  "properties": {
   "AcknowledgeType": {
    "example": 1,
    "type": "integer"
   },
   "Error": {
    "example": "Failed",
    "type": "string"
   }
  },
  "type": "object"
 }
```

```
      }
     },
     "security": [
      {
       "HMACAuth": []
      }
     ],
     "summary": "NOT IMPLEMENTED - Add calendar",
     "tags": [
      "not-implemented"
     ]
    }
   },
   "/api/door/getdoorstatus": {
    "get": {
     "description": "Returns the current status of each door.\n  Possible values:\n  - LOCKED\n  - UNLOCKED\n  - OPEN\n  - PERMANENT_UNLOCK\n  - PERMANENT_UNLOCK_OPEN\n  - NOT_CLOSED\n  - FORCED_OPEN\n",
     "produces": [
      "application/json"
     ],
     "responses": {
      "200": {
       "description": "Success",
       "examples": {
        "application/json": {
         "AcknowledgeType": 0,
         "DoorStatus": {
          "1": "LOCKED",
          "2": "LOCKED"
         },
         "Error": "None"
```

```
 }
},
"schema": {
 "properties": {
  "AcknowledgeType": {
   "example": 0,
   "type": "integer"
  },
  "DoorStatus": {
   "additionalProperties": {
    "description": "Door status",
    "enum": [
     "LOCKED",
     "UNLOCKED",
     "OPEN",
     "PERMANENT_UNLOCK",
     "PERMANENT_UNLOCK_OPEN",
     "NOT_CLOSED",
     "FORCED_OPEN"
    ],
    "type": "string"
   },
   "description": "Map of doorId -> door status",
   "type": "object"
  },
  "Error": {
   "example": "None",
   "type": "string"
  }
 },
 "required": [
```

```
      "AcknowledgeType",
      "DoorStatus",
      "Error"
    ],
    "type": "object"
  }
 },
 "401": {
  "description": "Unauthorized (invalid or missing HMAC headers)",
  "schema": {
    "properties": {
      "AcknowledgeType": {
        "example": 1,
        "type": "integer"
      },
      "Error": {
        "example": "Unauthorized",
        "type": "string"
      }
    },
    "required": [
      "AcknowledgeType",
      "Error"
    ],
    "type": "object"
  }
 },
 "500": {
  "description": "Failed",
  "schema": {
    "properties": {
```

```
        "AcknowledgeType": {
          "example": 1,
          "type": "integer"
        },
        "Error": {
          "example": "Failed",
          "type": "string"
        }
      },
      "required": [
        "AcknowledgeType",
        "Error"
      ],
      "type": "object"
    }
   }
  },
  "security": [
   {
    "HMACAuth": []
   }
  ],
  "summary": "Get door status",
  "tags": [
   "door"
  ]
 }
},
"/api/door/setoutput": {
 "post": {
  "consumes": [
```

```
      "application/json"
    ],
    "description": "Controls the door output.\n\n- `Set = true` \u2192 unlock\n- `Set = false` \u2192 lock\n- `Timed = true` \u2192 use pulse time\n- `Timed = false` \u2192 permanent change\n",
    "parameters": [
     {
      "in": "body",
      "name": "body",
      "required": true,
      "schema": {
       "example": {
        "Address": 1,
        "Set": true,
        "Timed": true
       },
       "properties": {
        "Address": {
         "description": "Reader/Door address",
         "example": 1,
         "type": "integer"
        },
        "Set": {
         "description": "Desired output state",
         "example": true,
         "type": "boolean"
        },
        "Timed": {
         "description": "Whether the change is timed",
         "example": true,
         "type": "boolean"
        }
```

```
    },
    "required": [
     "Address",
     "Set",
     "Timed"
    ],
    "type": "object"
   }
  }
 ],
 "produces": [
  "application/json"
 ],
 "responses": {
  "200": {
   "description": "Success",
   "schema": {
    "examples": {
     "application/json": {
      "AcknowledgeType": 0,
      "Error": "None"
     }
    },
    "properties": {
     "AcknowledgeType": {
      "example": 0,
      "type": "integer"
     },
     "Error": {
      "example": "None",
      "type": "string"
```

```
      }
    },
    "required": [
      "AcknowledgeType",
      "Error"
    ],
    "type": "object"
  }
},
"401": {
  "description": "Unauthorized (invalid or missing HMAC headers)",
  "schema": {
    "properties": {
      "AcknowledgeType": {
        "example": 1,
        "type": "integer"
      },
      "Error": {
        "example": "Unauthorized",
        "type": "string"
      }
    },
    "required": [
      "AcknowledgeType",
      "Error"
    ],
    "type": "object"
  }
},
"500": {
  "description": "Failed",
```

```
      "schema": {
       "properties": {
        "AcknowledgeType": {
         "example": 1,
         "type": "integer"
        },
        "Error": {
         "example": "Failed",
         "type": "string"
        }
       },
       "required": [
        "AcknowledgeType",
        "Error"
       ],
       "type": "object"
      }
     }
    },
    "security": [
     {
      "HMACAuth": []
     }
    ],
    "summary": "Set door output",
    "tags": [
     "door"
    ]
   }
  },
  "/api/door/setupdoor": {
```

```
"post": {
 "consumes": [
  "application/json"
 ],
 "description": "Verifies that the provided reader/door address is valid.\n",
 "parameters": [
  {
   "in": "body",
   "name": "body",
   "required": true,
   "schema": {
    "example": {
     "Address": 1
    },
    "properties": {
     "Address": {
      "description": "Reader/Door address to validate",
      "example": 1,
      "type": "integer"
     }
    },
    "required": [
     "Address"
    ],
    "type": "object"
   }
  }
 ],
 "produces": [
  "application/json"
 ],
```

```
"responses": {
 "200": {
  "description": "Success",
  "schema": {
   "examples": {
    "application/json": {
     "AcknowledgeType": 0,
     "Error": "None"
    }
   },
   "properties": {
    "AcknowledgeType": {
     "example": 0,
     "type": "integer"
    },
    "Error": {
     "example": "None",
     "type": "string"
    }
   },
   "required": [
    "AcknowledgeType",
    "Error"
   ],
   "type": "object"
  }
 },
 "401": {
  "description": "Unauthorized (invalid or missing HMAC headers)",
  "schema": {
   "properties": {
```

```
      "AcknowledgeType": {
        "example": 1,
        "type": "integer"
      },
      "Error": {
        "example": "Unauthorized",
        "type": "string"
      }
    },
    "required": [
      "AcknowledgeType",
      "Error"
    ],
    "type": "object"
  }
},
"500": {
  "description": "Failed",
  "schema": {
    "properties": {
      "AcknowledgeType": {
        "example": 1,
        "type": "integer"
      },
      "Error": {
        "example": "Failed",
        "type": "string"
      }
    },
    "required": [
      "AcknowledgeType",
```

```
          "Error"
         ],
         "type": "object"
        }
       }
      },
      "security": [
       {
        "HMACAuth": []
       }
      ],
      "summary": "Setup door",
      "tags": [
       "door"
      ]
     }
    },
    "/api/events": {
     "post": {
      "description": "Returns a list of events.\n\nEach event is represented as an array with fixed positions:\n1. Event type (string)\n2. Event timestamp (RFC 1123 date string)\n3. Event data (array)\n",
      "produces": [
       "application/json"
      ],
      "responses": {
       "200": {
        "description": "Success",
        "examples": {
         "application/json": {
          "AcknowledgeType": 0,
          "Error": "None",
```

```
      "Events": [
       [
        "AccessGranted",
        "Sat, 07 Feb 2026 12:47:27 GMT",
        [
         1,
         "44058 0029177"
        ]
       ]
      ]
     }
    },
    "schema": {
     "properties": {
      "AcknowledgeType": {
       "example": 0,
       "type": "integer"
      },
      "Error": {
       "example": "None",
       "type": "string"
      },
      "Events": {
       "description": "List of events",
       "items": {
        "description": "Event tuple:\n- index 0: Event type\n- index 1: Timestamp\n- index 2: Event data\n",
        "items": [
         {
          "description": "Event type.\n\nPossible values:\n- AccessGranted\n- AccessDenied\n- DoorOpenedTooLong\n- DoorOpenedForcibly\n- DoorOpenedRemotely\n- Tamper\n- TamperRecovered\n- UnknownEvent\n",
```

```
    "example": "AccessGranted",
    "type": "string"
   },
   {
    "description": "Event timestamp (RFC 1123)",
    "example": "Sat, 07 Feb 2026 12:47:27 GMT",
    "type": "string"
   },
   {
    "description": "Event-specific data",
    "example": [
     1,
     "44058 0029177"
    ],
    "items": {
     "oneOf": [
      {
       "type": "integer"
      },
      {
       "type": "string"
      }
     ]
    },
    "type": "array"
   }
  ],
  "maxItems": 3,
  "minItems": 3,
  "type": "array"
 },
```

```
      "type": "array"
     }
    },
    "required": [
     "AcknowledgeType",
     "Events",
     "Error"
    ],
    "type": "object"
   }
  },
  "401": {
   "description": "Unauthorized",
   "schema": {
    "properties": {
     "AcknowledgeType": {
      "example": 1,
      "type": "integer"
     },
     "Error": {
      "example": "Unauthorized",
      "type": "string"
     }
    },
    "type": "object"
   }
  },
  "500": {
   "description": "Failed",
   "schema": {
    "properties": {
```

```
        "AcknowledgeType": {
          "example": 1,
          "type": "integer"
        },
        "Error": {
          "example": "Failed",
          "type": "string"
        }
      },
      "type": "object"
    }
   }
  },
  "security": [
   {
    "HMACAuth": []
   }
  ],
  "summary": "Get events",
  "tags": [
   "root"
  ]
 }
},
"/api/keepalive": {
 "get": {
  "description": "Health/keepalive endpoint.",
  "produces": [
   "application/json"
  ],
  "responses": {
```

```
"200": {
 "description": "Success",
 "examples": {
  "application/json": {
   "AcknowledgeType": 0,
   "Error": "None"
  }
 },
 "schema": {
  "properties": {
   "AcknowledgeType": {
    "example": 0,
    "type": "integer"
   },
   "Error": {
    "example": "None",
    "type": "string"
   }
  },
  "required": [
   "AcknowledgeType",
   "Error"
  ],
  "type": "object"
 }
},
"500": {
 "description": "Failed",
 "examples": {
  "application/json": {
   "AcknowledgeType": 1,
```

```
          "Error": "Failed"
        }
      },
      "schema": {
        "properties": {
          "AcknowledgeType": {
            "example": 1,
            "type": "integer"
          },
          "Error": {
            "example": "Failed",
            "type": "string"
          }
        },
        "required": [
          "AcknowledgeType",
          "Error"
        ],
        "type": "object"
      }
    }
  },
  "summary": "Keepalive",
  "tags": [
    "root"
  ]
 }
},
"/api/login": {
 "post": {
```

```
    "description": "Authenticates a user via HTTP Basic Authentication and generates\na new API key + secret for HMAC-based requests.\n",
    "produces": [
      "application/json"
    ],
    "responses": {
      "200": {
        "description": "API key generated successfully",
        "schema": {
          "properties": {
            "Error": {
              "example": "None",
              "type": "string"
            },
            "KeyId": {
              "example": "a3f9c2e1b4d8...",
              "type": "string"
            },
            "Secret": {
              "description": "Hex-encoded secret used for HMAC signing",
              "example": "9f2a1c8e...",
              "type": "string"
            }
          },
          "type": "object"
        }
      },
      "401": {
        "description": "Unauthorized (invalid username/password)",
        "schema": {
          "properties": {
```

```
      "AcknowledgeType": {
        "example": 1,
        "type": "integer"
      },
      "Error": {
        "example": "Unauthorized",
        "type": "string"
      }
    },
    "type": "object"
  }
 },
 "500": {
  "description": "Failed to generate API key",
  "schema": {
    "properties": {
      "AcknowledgeType": {
        "example": 1,
        "type": "integer"
      },
      "Error": {
        "example": "Failed",
        "type": "string"
      }
    },
    "type": "object"
  }
 }
},
"security": [
 {
```

```
      "BasicAuth": []
     }
    ],
    "summary": "Generate HMAC API key",
    "tags": [
     "auth"
    ]
   }
  },
 "/api/restart": {
   "post": {
    "description": "Restarts the MeritAccess device.\n",
    "parameters": [
     {
      "in": "body",
      "name": "body",
      "required": false,
      "schema": {
       "type": "object"
      }
     }
    ],
    "produces": [
     "application/json"
    ],
    "responses": {
     "200": {
      "description": "Success",
      "examples": {
       "application/json": {
        "AcknowledgeType": 0,
```

```
      "Error": "None"
     }
    },
    "schema": {
     "properties": {
      "AcknowledgeType": {
       "example": 0,
       "type": "integer"
      },
      "Error": {
       "example": "None",
       "type": "string"
      }
     },
     "required": [
      "AcknowledgeType",
      "Error"
     ],
     "type": "object"
    }
   },
   "401": {
    "description": "Unauthorized (invalid or missing HMAC headers)",
    "schema": {
     "properties": {
      "AcknowledgeType": {
       "example": 1,
       "type": "integer"
      },
      "Error": {
       "example": "Unauthorized",
```

```
      "type": "string"
     }
    },
    "required": [
     "AcknowledgeType",
     "Error"
    ],
    "type": "object"
   }
  },
  "500": {
   "description": "Failed",
   "examples": {
    "application/json": {
     "AcknowledgeType": 1,
     "Error": "Failed"
    }
   },
   "schema": {
    "properties": {
     "AcknowledgeType": {
      "example": 1,
      "type": "integer"
     },
     "Error": {
      "example": "Failed",
      "type": "string"
     }
    },
    "required": [
     "AcknowledgeType",
```

```
            "Error"
          ],
          "type": "object"
        }
      }
    },
    "security": [
     {
       "HMACAuth": []
     }
    ],
    "summary": "Restart device",
    "tags": [
     "root"
    ]
   }
  }
 },
 "schemes": [
  "https"
 ],
 "securityDefinitions": {
  "BasicAuth": {
   "type": "basic"
  },
  "HMACAuth": {
   "description": "HMAC authentication.\n\nRequired headers:\n- X-Sign-Alg\n- X-Key-Id\n- X-Timestamp\n- X-Nonce\n- X-Signature",
   "in": "header",
   "name": "X-Signature",
   "type": "apiKey"
```

```
  }
 },
 "swagger": "2.0",
 "tags": [
  {
   "description": "Core system endpoints",
   "name": "root"
  },
  {
   "description": "Authentication",
   "name": "auth"
  },
  {
   "description": "Door control",
   "name": "door"
  },
  {
   "description": "Not implemented yet",
   "name": "not-implemented"
  }
 ]
}
```