



ALLEGION™

PIONEERING SAFETY™

What are the json files? More Postman Use

August 11th, 2025

What is a door file?

- Door files are json structured text documents that define the behavior of a lock. At a minimum it contains:
 - A “db” section, which can be nearly empty (a few empty sub sections)
 - A “schedules” section with at least one schedule (recommended 24/7)
 - “holidays” and “unlock” sections, which can be empty; do not apply to control locks. On NDEB and LEB, can be used to keep a door unlocked 9-5
 - A “Config” section which holds various configurable settings for the lock
 - A “nxtDbVerTS”, which should be a timestamp of the current database, but can be anything (it will simply be passed back by the lock)
 - See “Minimal doorfile.json”

What is a door file?

- The full specification of the door file is in “ENGAGE_JSON_Data_Structures - 2.18.pdf”
- This specification also contains data structures for the door config file, and door audit file. Door files are sent to the door via:
 - **BLE from Gateways in near real time communication over IP**
 - BLE from mobile app
 - Wi-Fi call in (NDEB, LEB, CTEB only)

What is a door config file?

- This is sent by the door in response for a request for config data (params) from the Gateway (or BLE mobile app or Wi-Fi)
- Contains various data describing the lock and its current configuration
- See “sample config file.json”

What is a door file's config section?

- Config can also be set via door file in a section titled “config”
- Two important tags:
 - “**RTCTime**” – sets the current clock of the lock in YYYYMMDDhhmmss format
 - “**invCrdAudEn**” – determines whether to send back transactions from unknown card events. Usually should be set to “T” (defaults to “F”)
- See “Config update doorfile.json”
- Note that tags which are omitted from a doorfile config section retain their preexisting value

What is the usrRcrd section of a door file?

- This section contains the card database for all cards (or incremental changes)
- Typical usage is to set “deleteAll” to true under the “db”->“usrRcrd” section
- Then populate all cards that should be allowed via the “add” array
- See “Doorfile with one card.json”
 - The encoding of the PrimeCR field holds the raw data for the credential, encrypted. See “Schlage ENGAGE Credential Sort and Encryption-1.34.pdf” for more details (we will go over this later in training)
- Additionally, after the db section, there is a property “nxtDbVerTS” – This is not strictly necessary, but should be set to a timestamp of when the doorfile was created. This allows the lock to report when its information is current, in the case of incremental updates to the door.

What is an audit file?

- Contains a record of all audits at the door (aka events)
- Each audit record contains an “event” tag and a “time” tag
 - Time is just the YYYYMMDDhhmmss when the audit occurred
 - The event tag is broken up into two sections
 - The first two bytes (four hex characters) are the event code (see “ENGAGE - Audits - 0.22.xlsm” tab DbmAuditEvents)
 - The latter two bytes are arbitrary data, dependent on the event code
- See “sample audits.json”

Starting to experiment in Postman

- Let's try to get a list of all devices linked to this gateway via a call to “{{gatewayIP}}/edgeDevices/linkList” (**Set Postman Variables**)
- 401 – Unauthorized! Oh no!
 - Each gateway needs to have a user/password set. Typically:
 - Randomly generated by the gateway soon after enrollment via GET
 - Then confirmed from the PACs software via PUT
 - PACs software must keep track of these
 - These per gateway values are tracked in addition to the site Encryption key
- Demonstrate the process (Good time to show postman Docs)
 - A GET then a PUT for “{{gatewayIP}}/gateway/newCredentials”
 - Auth scheme is a bit “quirky”
 - Update environment variables

Starting to experiment in Postman - 2

- Now we can list the linked devices – just one lock
- Let's ensure communication is working between gateway and lock?
 - Ideas on how to do this simply?
- Turning the inside handle is an easy way to generate audits
 - This will generate a request for exit
 - Then perform a GET “/edgeDevices/audits”
 - Note this can be done for all devices on a gateway, or a particular one
 - Show the slightly different URLs and postman variables
 - Should always delete audits after they have been retrieved and stored successfully on the host system

Starting to experiment in Postman - 3

- We can pulse a lock to momentary unlock
- See “momentary unlock.json” and the PUT `edgedevices/{lockID}/lockcontrol` call

Starting to experiment in Postman - 4

- Let's try swiping a random card and checking audits
 - Nothing appears! Why?
 - We need to enable “invCrdAudEn” to be “T”
- Default settings do not include failed access attempts
 - You can create your own set of defaults for a site
 - Failed access attempts include card data and can be useful for debugging
- Update lock config settings using sample json to include access denied and try again
 - Use “Config update doorfile.json”, (update **RTCTime** to current time)
 - PUT {{gatewayIP}}/edgeDevices/{{linkId}}/database
 - (Good time to show audit excel document and card encoding document “ENGAGE - Audits - 0.22.xlsm)

Schedules, auto-unlocks, and holidays

- Schedules are defined time periods when a card can be assigned access (Does not apply to Control locks)
 - As shown in “minimal doorfile.json”, there must always be at least one schedule, the first, which should be set to 24/7.
 - After that, there can be 15 more schedules, set to things like M-F 8am-6pm, etc
- Auto-unlocks
 - These are up to 32 weekly moments (repeating) when the door is set to go into a specified mode.
 - Typically, used in pairs. Sometimes, a single “secure” event is set for midnight to ensure the door is locked (in case it had been toggled
- Holidays
 - These are up to 32(16?) specific durations when the door is set to go into a specified mode
 - They override auto-unlocks, and are typically used when an exception to a normally passage door is required (e.g. on a holiday when office is closed)

Schedules, auto-unlocks, and holidays

- Examine the “M-F 8-6 unlock Doorfile with config update.json” doorfile
- Upload it to the lock
 - Your lock should unlock, assuming it is within the time specified by the auto-unlock events
- Note the holiday set for labor day
 - What do you think would happen on Labor day (a Monday)
 - The lock would not unlock during that day
- Auto-unlocks can also be configured for “first-person in”, where it won’t initially unlock until at least one valid access is presented (and then it will remain unlocked for the duration – for instance if there is a snow day and people are late for work, the door won’t be in passage mode while no one is in the office)

Monitoring the processing of a door file

- Examine the “JSON 50 user database.json” doorfile
 - Note it is in compact, rather than human readable whitespaced format
 - This can help speed processing
- Upload it to the lock
 - Then check GET
`{{gatewayIP}}/edgeDevices/{{linkId}}/dbDownloadStatus`
 - Notice it takes time for the lock to download and then process the doorfile
 - You cannot send another doorfile while one is processing
 - For huge files (e.g. 5000 users) this can take ~10 minutes