

**Goji server API V 0.0.18**

Glenn Widener

March 11, 2015, end of sprint 18, preliminary alpha2

Matches git describe tag+: v0.2.0-101-g9a935d1

Items marked TBD are not yet implemented as of the above tag.

Marked changes are relative to v0.0.14

**Overview**

API spec for Goji server, for alpha release.

Requests use standard get/post parameters, although the server can also accept ContentType=application/json requests with a 1-level json hash.

Content-type: application/x-www-form-urlencoded must be used to send binary image data (as BASE-64). Proposal is to switch this to Content-Type: multipart/form-data to support unencoded binary image data due to client encoder problems. Text is Latin-1 (8-bit ISO8859-1) unless noted.

Responses are JSON. Date/time values are represented as UTC in the ISO8601 format in both requests and responses, at least millisecond accuracy, e.g.: "2012-04-23T18:25:43.511Z". A null or non-"true" boolean value is to be interpreted as false ("true" is always lower-case). Null/missing string values are equivalent to the empty string. Except for PUT /keys, a missing/empty request parameter means do not change.

All reply data payloads (except a simple status: 200) include the current "server\_time ". Post-alpha, all request payloads should also include "time"; this will be used for security. Note that field order in reply payloads is NOT guaranteed, and order in requests does not matter.

Id's in request parameters may be zero, which means the same as not present/null/undefined (which is an error if the parameter is required).

An authtoken must be present on all app-originating requests to authenticate the requester (with the exception of the GET /authtoken call which presents an email and password, and as noted below on the /users endpoint during account creation). Authentication from a lock is TBD post- Alpha; for Alpha, an authtoken will not be required on endpoints that may be accessed from a lock (marked "from lock" below).

Authtoken and all params to GET /authtoken except device\_type should be passed in the header. user+pass are accepted using HTTP Basic authentication, authtoken as "Authorization: Token token=123123123". Other params are passed as X-Goji-<param-name> (note "-", not "\_" in "param-name"; some servers have improperly restricted header name characters).

Keys are RSA 1024 bit. Signing is with SHA1 hash, PKCS 1 padding. All binary data (e.g. signatures and keys) are encoded in JSON as base64 strings. Transmitted keys are public unless named "private".

**API**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| resource | verb | url | params\* | description |
| authtoken | GET | /authtoken | email, password,  device\_type (S13)  device\_model (S14)  os\_version (S14)  app\_version (S14)  optional:  device\_id, ua\_token  need\_keypair (S14) | return the authtoken (and the user+device id's, and UD keypair if need\_keypair). This is a singleton; there is only 1 authtoken for a given email+device\_id.  Beta: replace POST /devices, create and associate device with the authenticated user, sending device confirmation on new device. Resend new device confirmation if no device\_id. Else return 409 if device not yet confirmed for user. Update device+version info (required) ua\_token (omitted when unavailable) if authenticated. |
| authtoken | DELETE | /authtoken/<authtoken> |  | logout (eventually deleting token) |
| keys | GET  GET | /keys  /keys/id (TBD) |  | The keys owned by the authenticated user. Each key Includes whether they are admin for the lock, and the full name of the lock owner. /id restricts the returned data to a single key. |
| keys | POST | /keys | lock\_id, email, key\_data  (S5: deprecated the key\_data sub-key, accept flat POST parameters except time\_constraints) | Create a new key for the specified lock and user. Returns a new key in same format as an array entry for keys in the lock. if a param is not present, then the value in the database is unchanged from the default.  authenticated user must be lock owner/admin, or owner if admin true is supplied.  data: is\_fob, notify\_owner, notify\_locked, notify\_unlocked, notify\_denied, auto\_unlock, start\_date, end\_date, time\_constraints array (except valid\_days\_mask), admin (new S5) |
| keys | PUT | /keys/id (S6) | All POST key parameters except id's, email, revoked  <or> resend=true | Update the specified key. Returns the new key in same format as POST /keys. If any of \*date or time\_constraints is given, missing time or day values are set to empty, and a new id is returned (original key is revoked without notification). Otherwise, If a param is not present, then the value is unchanged.  Authenticated user must be lock owner/admin, or owner if key owner is admin or admin true is supplied.  If resend=true (S10), resend invitation email, with new temporary password and timeout if expired. Other fields ignored. Unchanged key is returned. |
| keys | DELETE | /keys/id |  | revoke a key (mark it revoked)  Authenticated user must be lock owner/admin, or owner if key owner is admin. |
| locks | GET  GET | /locks  /locks/id | owner\_only | The locks that the authenticated user owns, or for which they are an admin (if owner\_only=false/absent). (owner\_only TBD). /id restricts the returned data to a single lock. |
| locks | GET | /locks/credentials/<id>  /locks/credentials/serial/<lock\_serial> |  | The credentials and key list to validate the keys for the specified lock. |
| locks | POST  (NEW S4) | /locks  (from lock) | user\_id, lock\_serial (all required)  orientation, bolt\_state, name, internal\_version, external\_version, internal\_hw\_version, external\_hw\_version, bluetooth\_address, (all optional) | Add a lock.  user\_id is lock owner. Must be authenticated user (TBD)  Reply: complete resulting lock record including new lock id. |
| locks | PUT  (NEW S3)  PUT  (NEW S4) | /locks/serial/lock\_serial  (from lock)  /locks/id (lock\_id) | name, orientation, bolt\_state, image, bluetooth\_address,  battery\_level, battery\_state, internal\_version, external\_version, auto\_unlock\_owner, auto\_unlock\_others (all optional) | Change lock data. lock\_serial is the 96-bit hardware ID (for lock access, lock\_id for app)  bolt\_state= "locked", "unlocked", "failed"  orientation="left", "right"  battery\_level = 0-100  battery\_state = "low", "ok"  replacement of image not supported.  Validation: lock user\_id is authenticated user  Reply: complete resulting lock record, including required\_internal/external\_version, internal\_url, external\_url (and for alpha) internal/external\_ip, internal/external\_host, internal/external\_path (sans version string). \*\_url may contain "%h" which is to be replaced with hardware major version string (\*\_hw\_version) to download required\_\*\_version firmware (hardcoded "v2" for alpha.  reboot and debug\_log (S10) may be set in the server reply, they trigger lock actions. They are reset automatically on the server after the first PUT /locks/serlal request. |
| locks | DELETE  (NEW S11) | /locks/id |  | decommission a lock, mark as decommissioned, for future archival. Must be sysadmin or lock owner. |
| users | POST  (NEW S3) | /users | email | Create a new unconfirmed user account. Email must not exist and be confirmed, or 409 is returned. No authentication required. |
| users | GET  (NEW S3) | /users/email | password (optional) | Get the specified user's account data. Account must exist, or 404 is returned. If account has temporary password, return 200, Else email must be the authenticated user, or 401 is returned. Else return full account payload.  If password is supplied, no authentication token required, 401 if password incorrect, else 200 |
| users | PUT  (NEW S3, complete S7) | /users/<email> | first\_name, last\_name,  old\_password (S7),  password,  resend (S8), phone,  reset\_password (S9), confirmation\_token, ua\_token  device\_type  device\_model  os\_version  app\_version | Update the specified user's account data. If account is confirmed, email must be the authenticated user. If reset\_password\_token or old\_password is supplied, password is changed to password param. If resend=true: no auth required, the reset-password/account-create email with token is resent, If account is confirmed/not. If account is unconfirmed and either old\_password (receive access, temporary password) OR confirmation\_token (create account from app), password and account info are updated. If confirmation\_token, device\_\*, \*\_version are required to create device. |
| lock-user relation | PUT | /locks\_users/lock\_id | user\_id  admin | Associate a user with a lock, i.e. add/remove their admin privilege. Fields not supplied are unchanged. Authenticated user must be lock owner/admin |
| notifications | GET (removed) | /notifications |  | Get the notifications for the authenticated user not previously read |
| notifications | PUT (removed) | /notifications/id |  | Mark notification as read |
| events | POST | /events  (from lock) | lock\_id/lock\_serial, key\_id  event\_type  event\_time  string\_value  int\_value  bolt\_state | create a new event record for the specified key/lock. Server generates notifications. See Alpha Events and Notifications v10 spec.  event\_type= lock, unlock, wifi, battery, proximity, key\_shared, key\_revoked, key\_expired  string\_value= low, ok(battery), up, down(wifi), success, hardware\_failure, expired, use\_exceeded, outside\_time\_day(access)  int\_value=1-100(battery), ErrorEventCode (error\_notify\_\*)  bolt\_state = failed, locked, unlocked (failed" means "unknown" - can't determine the state)  One of lock\_id or lock\_serial required except for key\_\* events. |
| events | GET | /events  /events/id | lock\_id (optional TBD) | Lock owner or admin gets an array of all events for the lock (TBD), or for all locks they are owner/admin for, or one event by event id. (S18) |
| pictures | POST | /pictures  (from lock) | lock\_id/lock\_serial, taken\_at,  picture[:data],  picture[:original\_filename],  picture[:content\_type] | upload a picture for the specified lock, an attachment of an image file. |
| devices | POST | /devices/  (deprecated) | ua\_token  device\_type (new S5) | creates a device association with the authenticated user and registers it with AWS for push notifications\*. device\_type = "android", "ios"  Returns device\_id = id.  Triggers device validation sequence if not the first device for the account. This request rolled into GET /authtoken for beta.  \*note that if a device token for an existing user is created via POST, it will be removed (see delete) and recreated with the user indicated in the create call |
| devices | PUT | /devices/<device\_id>  (S13) | ua\_token, confirmation\_token  device\_model (S14)  os\_version (S14)  app\_version (S14) | update device info (for all users) |
| devices | DELETE | /devices/<device\_id>  (/devices/ua\_token/<ua\_token> for alpha 1) | none | Record user logout:remove user\_device.authenticated\_at. Deletes device\_id in alpha 1. |
| user\_devices | DELETE | /user\_devices/<user\_device\_id> | none | deletes the specified device association with the authenticated user, with all related records. |
| firmware | POST | /firmware (S7) | version, description, for\_external, file (required), description (optional) | Make a firmware download available to select for internal/external lock board. File is the downloadable firmware file. (TBD post-alpha: hardware\_version, "v2" for alpha) |
| log | POST | /logs (from lock and app) | device\_id, or lock\_id /lock\_serial,  string source,  binary data, fault\_time, (required),  string data\_type, string fault\_type, (optional) | (S9) Log device/lock history and state. If source="lock"/null, device\_id is optional, meaning a device paired with the lock at the time, and vice-versa: if source="ios" | "android", lock\_id/lock\_serial is optional.  If fault\_type is null, log is not due to a fault on the sending device, fault\_time is end of log. data\_type="text"|"binary"|null=binary. Data size limit 1MB? (TBD). Binary and text are application/x-www-form-urlencoded, base64 encoded for alpha. Both will be multipart/form-data for beta. |
| log | GET | /logs/serial/<lock\_serial>  /logs/<lock\_id>  /logs/device\_id/<device\_id>  /logs/ua\_token/<ua\_token> | source, fault\_time (required) | lock needs to query if a particular log is already in the server. Returns 200 (no data) if present, else 404. |

\*required unless otherwise indicated

**Responses**

The following is a sample of JSON data returned from each API call as actually generated. Arrays can contain 0-N hashes. Contents may be revised to add/delete database fields as needed. Endpoints not listed here simply return status: 200. "signature" values always immediately follow the sub-hash that they sign, otherwise, element order is not significant. Signatures sign the complete sub-hash, as transmitted, opening and closing curly brace inclusive.

Some of the more interesting response patterns:

request type:/locks/credentials GET

{

credentials :{

lock: lock\_id1,

users\_devices :{ "<user\_device\_id1>" : "<public\_key1>", …}

keys: [<key\_id1>, …],

servertime: "<time>",

expire: <seconds\_from\_servertime>

},

signature "<hash>"

}

Actual samples:

request type:Authtoken GET, no keypair

{

"authtoken": "GsULjFhg12dxYkWApYpg",

"user\_id": 405,

"user\_device\_id": 406,

"device\_id": 206

}

request type:Authtoken GET, new device

{

"device\_id": 229

}

request type:Authtoken GET, resend device confirmation

{

"device\_id": 229

}

request type:Devices PUT

{

"message": "Success",

"status": 200

}

request type:Devices POST

{

"server\_time": "2015-03-12T20:45:22.750474359Z",

"ua\_token": "64ce3cbfb716c0eb61adc2d6da1f2781e8b74d39b49c1e521108db9109b1be32",

"device\_type": "iOS",

"uuid": null,

"os\_version": null,

"app\_version": null,

"device\_model": null,

"id": 250,

"user\_device": {

"user\_id": 440,

"device\_id": 250,

"authenticated\_at": "2015-03-12T20:45:22.745Z",

"id": 454

}

}

request type:Devices DESTROY

{

"message": "Success",

"status": 200

}

request type:Events GET

[

{

"server\_time": "2015-03-12T20:46:55.536620069Z",

"key\_id": 367,

"lock\_id": 396,

"user\_id": 702,

"admin\_user\_id": null,

"picture\_id": 15,

"event\_type": "lock",

"string\_value": "success",

"int\_value": null,

"event\_time": "2015-03-12T20:46:54.278Z",

"bolt\_state": "locked",

"uuid": null,

"id": 861,

"picture\_url": "https://goji-server-test.s3.amazonaws.com/pictures/data/000/000/015/original/test.jpeg?AWSAccessKeyId=AKIAJ6OKLEYPV4YFHMIA&Expires=1426193225&Signature=rvD7ZOmMmK1O%2BhDIpX0yW1uZLa0%3D",

"lock\_name": "Front Door",

"key\_user\_id": 702,

"key\_user\_display\_name": "John Doe",

"user\_display\_name": "John Doe"

},

{

"server\_time": "2015-03-12T20:46:55.538796402Z",

"key\_id": 367,

"lock\_id": 396,

"user\_id": 702,

"admin\_user\_id": null,

"picture\_id": 15,

"event\_type": "unlock",

"string\_value": "success",

"int\_value": null,

"event\_time": "2015-03-12T20:46:54.278Z",

"bolt\_state": "unlocked",

"uuid": null,

"id": 862,

"picture\_url": "https://goji-server-test.s3.amazonaws.com/pictures/data/000/000/015/original/test.jpeg?AWSAccessKeyId=AKIAJ6OKLEYPV4YFHMIA&Expires=1426193225&Signature=rvD7ZOmMmK1O%2BhDIpX0yW1uZLa0%3D",

"lock\_name": "Front Door",

"key\_user\_id": 702,

"key\_user\_display\_name": "John Doe",

"user\_display\_name": "John Doe"

},

{

"server\_time": "2015-03-12T20:46:55.544248309Z",

"key\_id": 367,

"lock\_id": 396,

"user\_id": 702,

"admin\_user\_id": null,

"picture\_id": 15,

"event\_type": "lock",

"string\_value": "invalid\_key",

"int\_value": null,

"event\_time": "2015-03-12T20:46:54.278Z",

"bolt\_state": "unlocked",

"uuid": null,

"id": 863,

"picture\_url": "https://goji-server-test.s3.amazonaws.com/pictures/data/000/000/015/original/test.jpeg?AWSAccessKeyId=AKIAJ6OKLEYPV4YFHMIA&Expires=1426193225&Signature=rvD7ZOmMmK1O%2BhDIpX0yW1uZLa0%3D",

"lock\_name": "Front Door",

"key\_user\_id": 702,

"key\_user\_display\_name": "John Doe",

"user\_display\_name": "John Doe"

},

{

"server\_time": "2015-03-12T20:46:55.546261581Z",

"key\_id": null,

"lock\_id": 396,

"user\_id": null,

"admin\_user\_id": null,

"picture\_id": null,

"event\_type": "battery",

"string\_value": "low",

"int\_value": 0,

"event\_time": "2015-03-12T20:46:54.278Z",

"bolt\_state": null,

"uuid": null,

"id": 864,

"picture\_url": null,

"lock\_name": "Front Door"

},

{

"server\_time": "2015-03-12T20:46:55.546521135Z",

"key\_id": null,

"lock\_id": 397,

"user\_id": null,

"admin\_user\_id": null,

"picture\_id": null,

"event\_type": "wifi",

"string\_value": "down",

"int\_value": null,

"event\_time": "2015-03-12T20:46:54.278Z",

"bolt\_state": null,

"uuid": null,

"id": 865,

"picture\_url": null,

"lock\_name": "Front Door"

},

{

"server\_time": "2015-03-12T20:46:55.547696630Z",

"key\_id": null,

"lock\_id": 397,

"user\_id": null,

"admin\_user\_id": null,

"picture\_id": null,

"event\_type": "proximity",

"string\_value": "into",

"int\_value": null,

"event\_time": "2015-03-12T20:46:54.278Z",

"bolt\_state": null,

"uuid": null,

"id": 866,

"picture\_pending": "true",

"picture\_url": null,

"lock\_name": "Front Door"

},

{

"server\_time": "2015-03-12T20:46:55.551682788Z",

"key\_id": null,

"lock\_id": 396,

"user\_id": null,

"admin\_user\_id": null,

"picture\_id": 15,

"event\_type": "lock",

"string\_value": "success",

"int\_value": null,

"event\_time": "2015-03-12T20:46:54.278Z",

"bolt\_state": "locked",

"uuid": null,

"id": 867,

"picture\_url": "https://goji-server-test.s3.amazonaws.com/pictures/data/000/000/015/original/test.jpeg?AWSAccessKeyId=AKIAJ6OKLEYPV4YFHMIA&Expires=1426193225&Signature=rvD7ZOmMmK1O%2BhDIpX0yW1uZLa0%3D",

"lock\_name": "Front Door"

},

{

"server\_time": "2015-03-12T20:46:55.557105870Z",

"key\_id": null,

"lock\_id": 396,

"user\_id": null,

"admin\_user\_id": null,

"picture\_id": 15,

"event\_type": "unlock",

"string\_value": "hardware\_failure",

"int\_value": null,

"event\_time": "2015-03-12T20:46:54.278Z",

"bolt\_state": "locked",

"uuid": null,

"id": 868,

"picture\_url": "https://goji-server-test.s3.amazonaws.com/pictures/data/000/000/015/original/test.jpeg?AWSAccessKeyId=AKIAJ6OKLEYPV4YFHMIA&Expires=1426193225&Signature=rvD7ZOmMmK1O%2BhDIpX0yW1uZLa0%3D",

"lock\_name": "Front Door"

},

{

"server\_time": "2015-03-12T20:46:55.559202114Z",

"key\_id": null,

"lock\_id": 396,

"user\_id": null,

"admin\_user\_id": null,

"picture\_id": null,

"event\_type": "wifi",

"string\_value": "down",

"int\_value": null,

"event\_time": "2015-03-12T20:46:54.278Z",

"bolt\_state": "unlocked",

"uuid": null,

"id": 869,

"picture\_url": null,

"lock\_name": "Front Door"

},

{

"server\_time": "2015-03-12T20:46:55.559420507Z",

"key\_id": 367,

"lock\_id": 396,

"user\_id": 702,

"admin\_user\_id": null,

"picture\_id": null,

"event\_type": "error\_notify\_sysadmin",

"string\_value": null,

"int\_value": 1,

"event\_time": "2015-03-12T20:46:54.278Z",

"bolt\_state": null,

"uuid": null,

"id": 870,

"picture\_url": null,

"lock\_name": "Front Door",

"key\_user\_id": 702,

"key\_user\_display\_name": "John Doe",

"user\_display\_name": "John Doe"

},

{

"server\_time": "2015-03-12T20:46:55.559613520Z",

"key\_id": 367,

"lock\_id": 396,

"user\_id": 702,

"admin\_user\_id": null,

"picture\_id": null,

"event\_type": "error\_notify\_owner\_admin",

"string\_value": null,

"int\_value": 2,

"event\_time": "2015-03-12T20:46:54.278Z",

"bolt\_state": null,

"uuid": null,

"id": 871,

"picture\_url": null,

"lock\_name": "Front Door",

"key\_user\_id": 702,

"key\_user\_display\_name": "John Doe",

"user\_display\_name": "John Doe"

},

{

"server\_time": "2015-03-12T20:46:55.560148048Z",

"key\_id": null,

"lock\_id": 396,

"user\_id": 702,

"admin\_user\_id": 703,

"picture\_id": null,

"event\_type": "admin\_shared",

"string\_value": null,

"int\_value": null,

"event\_time": "2015-03-12T20:46:54.243Z",

"bolt\_state": null,

"uuid": null,

"id": 860,

"picture\_url": null,

"lock\_name": "Front Door",

"user\_display\_name": "John Doe",

"admin\_display\_name": "John Doe"

},

{

"server\_time": "2015-03-12T20:46:55.561862132Z",

"key\_id": null,

"lock\_id": 396,

"user\_id": 702,

"admin\_user\_id": 704,

"picture\_id": null,

"event\_type": "admin\_shared",

"string\_value": null,

"int\_value": null,

"event\_time": "2015-03-12T20:46:54.212Z",

"bolt\_state": null,

"uuid": null,

"id": 859,

"picture\_url": null,

"lock\_name": "Front Door",

"user\_display\_name": "John Doe",

"admin\_display\_name": "John Doe"

},

{

"server\_time": "2015-03-12T20:46:55.563538688Z",

"key\_id": 367,

"lock\_id": 396,

"user\_id": 702,

"admin\_user\_id": null,

"picture\_id": null,

"event\_type": "key\_shared",

"string\_value": null,

"int\_value": null,

"event\_time": "2015-03-12T20:46:54.127Z",

"bolt\_state": null,

"uuid": null,

"id": 857,

"picture\_url": null,

"lock\_name": "Front Door",

"key\_user\_id": 702,

"key\_user\_display\_name": "John Doe",

"user\_display\_name": "John Doe"

}

]

request type:Keys POST

{

"key\_data": {

"id": 386,

"lock\_id": 412,

"user\_id": 749,

"start\_date": null,

"end\_date": null,

"time\_constraints": [

]

},

"key\_info": {

"server\_time": "2015-03-12T20:47:43.411395746Z",

"sharer\_user\_id": 747,

"revoked": null,

"replaced\_at": null,

"name": "Front Door",

"is\_fob": null,

"pending": true,

"use\_count": null,

"use\_limit": null,

"notify\_owner": null,

"auto\_unlock": null,

"uuid": null,

"id": 386,

"seq\_no": 2,

"bluetooth\_address": "ALEX-1234",

"last\_access": null,

"first\_name": "John",

"last\_name": "Doe",

"email": "glenn3@example.com",

"user\_display\_name": "John Doe",

"lock\_user": {

"admin": false,

"uuid": null,

"id": 250,

"note": "Legacy for GET /keys only, deprecated, use admin flag"

},

"admin": false,

"time\_constraints": [

]

}

}

request type:Keys GET

[

{

"key\_data": {

"id": 449,

"lock\_id": 455,

"user\_id": 871,

"start\_date": null,

"end\_date": null,

"time\_constraints": [

{

"valid\_days\_mask": 61,

"start\_offset": 3600,

"end\_offset": 7200

}

]

},

"signature": "QkD4ZgiT5BU7P+iRgfQjVImich52skFgq0e3bdQJ1A7UQpBUUqpRkewzI2VV\nEJFyGoXQ7DbFXlN0OSxpIiTEdyyTZwDD7n61WPD4L4w58U1kC8O8PT8NSmKd\nvWYqawZBvm4ld7kgUrIhqx/JX9dNrfXe9CVcja8IXe8ZWXY1Rc8=\n",

"key\_info": {

"server\_time": "2015-03-12T20:48:17.456031028Z",

"sharer\_user\_id": 871,

"revoked": null,

"replaced\_at": null,

"name": "Front Door",

"is\_fob": false,

"pending": false,

"use\_count": null,

"use\_limit": null,

"notify\_owner": true,

"auto\_unlock": null,

"uuid": null,

"id": 449,

"seq\_no": 2,

"bluetooth\_address": "ALEX-1234",

"last\_access": null,

"first\_name": "John",

"last\_name": "Doe",

"email": "glenn2@example.com",

"user\_display\_name": "John Doe",

"lock\_user": {

"admin": true,

"uuid": null,

"id": 302,

"note": "Legacy for GET /keys only, deprecated, use admin flag"

},

"auto\_generated": true,

"admin": true,

"time\_constraints": [

{

"monday": true,

"tuesday": true,

"wednesday": true,

"thursday": true,

"friday": false,

"saturday": true,

"sunday": null,

"id": 384

}

],

"lock\_owner\_display\_name": "John Doe",

"sharer\_display\_name": "John Doe"

}

}

]

request type:Keys PUT

{

"key\_data": {

"id": 478,

"lock\_id": 472,

"user\_id": 907,

"start\_date": null,

"end\_date": null,

"time\_constraints": [

{

"valid\_days\_mask": 0,

"start\_offset": null,

"end\_offset": null

}

]

},

"key\_info": {

"server\_time": "2015-03-12T20:48:27.324936619Z",

"sharer\_user\_id": 905,

"revoked": null,

"replaced\_at": null,

"name": "Front Door",

"is\_fob": true,

"pending": true,

"use\_count": 10,

"use\_limit": 100,

"notify\_owner": null,

"auto\_unlock": true,

"uuid": null,

"id": 478,

"seq\_no": 5,

"bluetooth\_address": "ALEX-1234",

"last\_access": null,

"first\_name": "John",

"last\_name": "Doe",

"email": "glennguest@example.com",

"user\_display\_name": "John Doe",

"lock\_user": {

"admin": false,

"uuid": null,

"id": 327,

"note": "Legacy for GET /keys only, deprecated, use admin flag"

},

"admin": false,

"time\_constraints": [

{

"monday": null,

"tuesday": null,

"wednesday": null,

"thursday": null,

"friday": null,

"saturday": null,

"sunday": null,

"id": 393

}

]

}

}

request type:Keys PUT

{

"key\_data": {

"id": 504,

"lock\_id": 482,

"user\_id": 937,

"start\_date": null,

"end\_date": null,

"time\_constraints": [

]

},

"key\_info": {

"server\_time": "2015-03-12T20:48:35.653867005Z",

"sharer\_user\_id": 935,

"revoked": null,

"replaced\_at": null,

"name": "Front Door",

"is\_fob": true,

"pending": true,

"use\_count": null,

"use\_limit": 100,

"notify\_owner": null,

"auto\_unlock": true,

"uuid": null,

"id": 504,

"seq\_no": 3,

"bluetooth\_address": "ALEX-1234",

"last\_access": null,

"first\_name": "John",

"last\_name": "Doe",

"email": "glennguest@example.com",

"user\_display\_name": "John Doe",

"lock\_user": {

"admin": true,

"uuid": null,

"id": 347,

"note": "Legacy for GET /keys only, deprecated, use admin flag"

},

"auto\_generated": true,

"admin": true,

"time\_constraints": [

]

}

}

request type:Locks POST

{

"server\_time": "2015-03-12T20:48:46.228733386Z",

"user\_id": 969,

"name": "my\_lock",

"bolt\_state": "unlocked",

"bluetooth\_address": "BLUETOOTH-NAME1",

"orientation": "left",

"commission\_date": null,

"decommission\_date": null,

"internal\_version": "0.5.5",

"external\_version": "0.5.5abc",

"required\_external\_version": "0.0.9T",

"required\_internal\_version": "0.0.9V",

"battery\_level": null,

"internal\_hw\_version": null,

"external\_hw\_version": null,

"battery\_state": null,

"lock\_serial": "17a70818b8ff88ac7eafae8ef2f49c3b",

"auto\_unlock\_owner": null,

"auto\_unlock\_others": null,

"reboot": false,

"debug\_log": false,

"uuid": null,

"new\_credentials": true,

"serial\_label": null,

"reported\_wifi\_status": "up",

"reported\_wifi\_time": null,

"last\_sync": null,

"id": 496,

"seq\_no": 1,

"last\_access": null,

"status": "unlocked"

}

request type:/locks/credentials GET

{

"credentials": {

"users\_devices": {

"985": "LS0tLS1CRUdJTiBQVUJMSUMgS0VZLS0tLS0KTUlHZk1BMEdDU3FHU0liM0RR\nRUJBUVVBQTRHTkFEQ0JpUUtCZ1FETjJkYy9BN0p2cUFzd3NaWmpFbTdiNDdh\negpFS2Z2NFNKRGtHcjRxWkNtUjFLUjA0VDkxNnJyZzhTME8vU2VVWkVOSzds\nL1pBa3BibHRpNFBOZFo3cUxOVEd6CnNIR2NveXltaTBoUVh2amFRREdWdElV\nYUxURUlxN25yS0E1RXJXYys3clpQZStHU1JNK0xObW9oMktlSHB6N2QKaHk0\nci9JaHp2bFpTVHNaUkVRSURBUUFCCi0tLS0tRU5EIFBVQkxJQyBLRVktLS0t\nLQo=\n",

"986": "LS0tLS1CRUdJTiBQVUJMSUMgS0VZLS0tLS0KTUlHZk1BMEdDU3FHU0liM0RR\nRUJBUVVBQTRHTkFEQ0JpUUtCZ1FETjJkYy9BN0p2cUFzd3NaWmpFbTdiNDdh\negpFS2Z2NFNKRGtHcjRxWkNtUjFLUjA0VDkxNnJyZzhTME8vU2VVWkVOSzds\nL1pBa3BibHRpNFBOZFo3cUxOVEd6CnNIR2NveXltaTBoUVh2amFRREdWdElV\nYUxURUlxN25yS0E1RXJXYys3clpQZStHU1JNK0xObW9oMktlSHB6N2QKaHk0\nci9JaHp2bFpTVHNaUkVRSURBUUFCCi0tLS0tRU5EIFBVQkxJQyBLRVktLS0t\nLQo=\n",

"987": "LS0tLS1CRUdJTiBQVUJMSUMgS0VZLS0tLS0KTUlHZk1BMEdDU3FHU0liM0RR\nRUJBUVVBQTRHTkFEQ0JpUUtCZ1FETjJkYy9BN0p2cUFzd3NaWmpFbTdiNDdh\negpFS2Z2NFNKRGtHcjRxWkNtUjFLUjA0VDkxNnJyZzhTME8vU2VVWkVOSzds\nL1pBa3BibHRpNFBOZFo3cUxOVEd6CnNIR2NveXltaTBoUVh2amFRREdWdElV\nYUxURUlxN25yS0E1RXJXYys3clpQZStHU1JNK0xObW9oMktlSHB6N2QKaHk0\nci9JaHp2bFpTVHNaUkVRSURBUUFCCi0tLS0tRU5EIFBVQkxJQyBLRVktLS0t\nLQo=\n",

"988": "LS0tLS1CRUdJTiBQVUJMSUMgS0VZLS0tLS0KTUlHZk1BMEdDU3FHU0liM0RR\nRUJBUVVBQTRHTkFEQ0JpUUtCZ1FETjJkYy9BN0p2cUFzd3NaWmpFbTdiNDdh\negpFS2Z2NFNKRGtHcjRxWkNtUjFLUjA0VDkxNnJyZzhTME8vU2VVWkVOSzds\nL1pBa3BibHRpNFBOZFo3cUxOVEd6CnNIR2NveXltaTBoUVh2amFRREdWdElV\nYUxURUlxN25yS0E1RXJXYys3clpQZStHU1JNK0xObW9oMktlSHB6N2QKaHk0\nci9JaHp2bFpTVHNaUkVRSURBUUFCCi0tLS0tRU5EIFBVQkxJQyBLRVktLS0t\nLQo=\n",

"989": "LS0tLS1CRUdJTiBQVUJMSUMgS0VZLS0tLS0KTUlHZk1BMEdDU3FHU0liM0RR\nRUJBUVVBQTRHTkFEQ0JpUUtCZ1FETjJkYy9BN0p2cUFzd3NaWmpFbTdiNDdh\negpFS2Z2NFNKRGtHcjRxWkNtUjFLUjA0VDkxNnJyZzhTME8vU2VVWkVOSzds\nL1pBa3BibHRpNFBOZFo3cUxOVEd6CnNIR2NveXltaTBoUVh2amFRREdWdElV\nYUxURUlxN25yS0E1RXJXYys3clpQZStHU1JNK0xObW9oMktlSHB6N2QKaHk0\nci9JaHp2bFpTVHNaUkVRSURBUUFCCi0tLS0tRU5EIFBVQkxJQyBLRVktLS0t\nLQo=\n"

},

"keys": [

545,

546

],

"server\_time": "2015-03-12T20:48:52.933831248Z",

"expire": 60,

"lock": 506

},

"signature": "RYJEdwECn6rY+XfpGSpv2qpJTs4F0WRPONrbaKyazXA5XQx8EXjoX/dq0/FB\nGqTpcpc5mt+ak3MtyOzQwFapKPRIcGSumxk2S59ckoOAgadekc3LCYP7Ykty\n1LsvQXiSi3E2i4Os5PbieoBcaU13TUlUZH92UE2fohGdNfKNLwc=\n"

}

request type:Locks GET

[

{

"server\_time": "2015-03-12T20:49:08.961577316Z",

"user\_id": 1005,

"name": "Front Door",

"bolt\_state": "unlocked",

"bluetooth\_address": "ALEX-1234",

"orientation": "left",

"commission\_date": "2015-03-12T20:49:08.876Z",

"decommission\_date": null,

"internal\_version": "0.0.9V",

"external\_version": "0.0.9T",

"required\_external\_version": "0.0.9T",

"required\_internal\_version": "0.0.9V",

"battery\_level": null,

"internal\_hw\_version": null,

"external\_hw\_version": null,

"battery\_state": null,

"lock\_serial": "4ae83da1b783074e0f0f04d7e31d0cbf",

"auto\_unlock\_owner": null,

"auto\_unlock\_others": null,

"reboot": false,

"debug\_log": false,

"uuid": null,

"new\_credentials": true,

"serial\_label": null,

"reported\_wifi\_status": "up",

"reported\_wifi\_time": null,

"last\_sync": null,

"id": 519,

"seq\_no": 1,

"last\_access": "2015-03-12T20:49:08.925Z",

"status": "unlocked",

"keys": [

{

"key\_data": {

"id": 572,

"lock\_id": 519,

"user\_id": 1005,

"start\_date": null,

"end\_date": null,

"time\_constraints": [

{

"valid\_days\_mask": 61,

"start\_offset": 3600,

"end\_offset": 7200

}

]

},

"key\_info": {

"sharer\_user\_id": 1005,

"revoked": null,

"replaced\_at": null,

"name": "Front Door",

"is\_fob": false,

"pending": false,

"use\_count": null,

"use\_limit": null,

"notify\_owner": true,

"auto\_unlock": null,

"uuid": null,

"id": 572,

"seq\_no": 2,

"bluetooth\_address": "ALEX-1234",

"last\_access": "2015-03-12T20:49:08.925Z",

"first\_name": "John",

"last\_name": "Doe",

"email": "jdoe@emmoco.com",

"user\_display\_name": "John Doe",

"admin": false,

"time\_constraints": [

{

"monday": true,

"tuesday": true,

"wednesday": true,

"thursday": true,

"friday": false,

"saturday": true,

"sunday": null,

"id": 436

}

]

}

}

],

"event\_type": "unlock",

"event\_string\_value": "hardware\_failure",

"event\_bolt\_state": "failed",

"event\_user\_full\_name": "John Doe",

"event\_user\_id": 1005,

"event\_time": "2015-03-12T20:49:08.925Z"

}

]

request type:Locks PUT

{

"server\_time": "2015-03-12T20:49:49.572312044Z",

"user\_id": 1055,

"name": "Front Door",

"bolt\_state": "unlocked",

"bluetooth\_address": "ALEX-1234",

"orientation": "left",

"commission\_date": "2015-03-12T20:49:49.516Z",

"decommission\_date": null,

"internal\_version": "0.0.9V",

"external\_version": "0.0.9T",

"required\_external\_version": "0.0.9T",

"required\_internal\_version": "0.0.9V",

"battery\_level": null,

"internal\_hw\_version": null,

"external\_hw\_version": null,

"battery\_state": null,

"lock\_serial": "734b6c05c995b8fdff96deea2a990c21",

"auto\_unlock\_owner": false,

"auto\_unlock\_others": true,

"reboot": false,

"debug\_log": false,

"uuid": null,

"new\_credentials": true,

"serial\_label": null,

"reported\_wifi\_status": "up",

"reported\_wifi\_time": null,

"last\_sync": null,

"id": 561,

"seq\_no": 2,

"last\_access": null,

"image\_url": "https://goji-server-test.s3.amazonaws.com/locks/images/000/000/040/original/test.jpeg?AWSAccessKeyId=AKIAJ6OKLEYPV4YFHMIA&Expires=1414526492&Signature=jVkE4rTu0PEuKb%2BC7XSlP4vfQwY%3D",

(Note: image\_url Is disabled for alpha because it's unused and too long for firmware)

"status": "unlocked",

"external\_host": "s3.amazonaws.com",

"external\_path": "/goji-firmware/external/v2/",

"external\_ip": "54.231.244.8",

"internal\_url": "ftp://gate-kpepzd:sa4YjpxjxWIi@ftp.cloudgates.net/internal/v2/0.0.9V",

"internal\_host": "ftp.cloudgates.net",

"internal\_path": "/internal/v2/",

"internal\_user": "gate-kpepzd",

"internal\_pass": "sa4YjpxjxWIi",

"internal\_ip": "209.141.43.139"

}

request type:Logs POST

{

"message": "Success",

"status": 200

}

request type:Logs GET /logs/id

{

"message": "Success",

"status": 200

}

request type:UserDevices DESTROY

{

"message": "Success",

"status": 200

}

request type:UserDevices PUT

{

"message": "Success",

"status": 200

}

request type:Users GET

{

"server\_time": "2015-03-12T20:50:57.310732345Z",

"time\_zone": "Central Time (US & Canada)",

"uuid": null,

"id": 1127,

"account": {

"first\_name": "John",

"last\_name": "Doe",

"full\_name": "John Doe",

"email": "jdoe@emmoco.com",

"confirmed\_at": "2015-03-12T20:50:56.091Z",

"confirmation\_sent\_at": "2015-03-12T20:50:56.091Z",

"uuid": null,

"id": 1129

}

}

**Mobile Cache management**

It is assumed that the mobile app will cache data on locks, keys, users and notifications. The general mechanism for controlling this interface is a seq\_no that will be incremented by the server whenever any change has been made to the object and a reload is required. (implemented in server for locks and keys only.)

The mobile app should compare the seq\_no of an object to what it has in its cache and if they are the same, then no update is required. Using a GET on the seq\_nos provides a single quick call to determine if your local data requires updates.

**Images**

Images will be hosted on S3. When the app requests an object from the server if the object has an associated image, then an s3 url to that image will be returned. The url will be good for 5 minutes and it is assumed that the mobile device will download the image and cache it.

**Errors returned from api**

Http status codes are always returned with an error message, and the status code both in the header and as a "status" field in the JSON body. Successful replies have a data payload or simply status: 200. Here is an example of an error message body return from the api:

|  |
| --- |
| {"error":"Unauthorized","status":401} |

|  |  |  |
| --- | --- | --- |
| 200 | OK | Success! |
| 400 | Bad Request | The request was invalid. We (TBD) never return 400 in server code, you should only get this due to e.g. improperly formatted json. See 422. |
| 401 | Unauthorized | Authentication credentials were missing or incorrect. This will be returned if an auth token is not valid, if an invalid  email / password combination is used to request an auth token or an email or password is missing, or if an account confirmation/temporary password is invalid. |
| 403 | Forbidden | The request is understood, but it has been refused or access is not allowed. An accompanying error message will explain why. Retrying these errors will not typically succeed. Example: you tried to edit a lock that you did not own or have admin rights to, or old password is invalid on password change. |
| 404 | Not Found | The URI requested is invalid or the resource requested does not exist. This includes a parameter which does not exist, e.g. a lock id. |
| 409 | Conflict | The resource is in a conflicting state to the request. Here: resource already exists on a create request (e.g. new account email) |
| 422 | Invalid Data | Bad parameters, either missing or invalid. An accompanying error message will explain why. For example, if in commissioning a lock (not yet defined) a param was required and it wasn't included in the call. |
| 500 | Internal Server Error | Something is broken in the server |

**Notification strings**

Strings generated for push notifications on the server are documented in the Events and Notifications spreadsheet.

**To Do**

* API versioning in urls
* add list of all error messages