**RESTful API for Keylime (*v2*)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *tenant -> cloud verifier (CV)* | GET (retrieve) | PUT (replace/edit) | POST (new) | DELETE (remove) |
| /v2/agents/ | Get list of agents from CV |  |  |  |
| /v2/agents/*{agent\_id:UUID}* | Get status of agent agent\_id from CV |  | Add new agent agent\_id to CV  Requires JSON body:  ***{***  ***“v” : key,***  ***“ip” : ipaddr,***  ***“port” : int,***  ***“operational\_state” : int,***  ***“public\_key” : key,***  ***“tpm\_policy” : json,***  ***“vtpm\_policy” : json,***  ***“metadata” : json,***  ***“ima\_whitelist” : json,***  ***“accept\_tpm\_hash\_algs”: list,***  ***“accept\_tpm\_encryption\_algs”: list,***  ***“accept\_tpm\_signing\_algs”: list,***  ***}*** | Terminate agent agent\_id |
| /v2/agents/*{agent\_id:UUID}*/reactivate |  | Start agent agent\_id  *For an already bootstrapped agent\_id agent* |  |  |
| /v2/agents/*{agent\_id:UUID}*/stop |  | Stop cv polling on agent\_id, but don’t delete  *For an already started agent\_id* |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *verifier/tenant -> agent* | GET (retrieve) | PUT (replace/edit) | POST (new) | DELETE (remove) |
| /v2/keys/ |  |  |  |  |
| /v2/keys/pubkey | Retrieves agent’s public key |  |  |  |
| /v2/keys/vkey |  |  | Send v\_key to agent  Requires JSON body:  ***{***  ***“encrypted\_key”: key,***  ***}*** |  |
| /v2/keys/ukey |  |  | Send u\_key to agent (with optional payload)  Requires JSON body:  ***{***  ***“auth\_tag” : hmac,***  ***“encrypted \_key”: key,***  ***“payload”: b64,*** (*opt*)  ***}*** |  |
| /v2/keys/verify  ex.)  */v2/keys/verify?challenge=#* | Get confirmation of bootstrap key derivation  Requires query parameters:  ***challenge : int*** |  |  |  |
| /v2/quotes/ |  |  |  |  |
| /v2/quotes/integrity  ex.)  */v2/quotes/integrity?nonce=#&mask=#&vmask=#&partial=#* | Get integrity quote from agent  Required parameters:  ***nonce : int***  ***mask : bitmask***  ***vmask : bitmask***  ***partial : bool*** |  |  |  |
| /v2/quotes/identity  ex.)  */v2/quotes/identity?nonce=#* | Get identity quote from agent  Requires query parameters:  ***nonce : int*** |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *verifier/agent/tenant -> registrar* | GET (retrieve) | PUT (replace/edit) | POST (new) | DELETE (remove) |
| /v2/agents/ | Get list of registered agents |  |  |  |
| /v2/agents/{agent\_id:UUID} | Get AIK of agent agent\_id  *Typically used by verifier/tenant* |  | Add agent agent\_id to registrar  Requires JSON body:  ***{***  ***“ek” : key,***  ***“ekcert” : cert,***  ***“aik” : key,***  ***“tpm\_version”: TPM version,***  ***“aik\_name” : key name,*** (*tpm2*)  ***“ek\_tpm” : TPM-format key*** (*tpm2*)  ***}*** | Remove agent agent\_id from registrar |
| /v2/agents/{agent\_id:UUID}/activate |  | Activate physical agent agent\_id  Requires JSON body:  ***{***  ***“auth\_tag” : hmac,***  ***}*** |  |  |
| /v2/agents/{agent\_id:UUID}/vactivate |  | Activate virtual agent agent\_id  Requires JSON body:  ***{***  ***“deepquote” : b64,***  ***}*** |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Tenant webapp* | GET (retrieve) | PUT (replace/edit) | POST (new) | DELETE (remove) |
| /v2/agents/ | Get ordered list of agents (by state) |  |  |  |
| /v2/agents/{agent\_id:UUID} | Get details of specified agent\_id | Start agent agent\_id  *For an already bootstrapped agent\_id agent* | Add new agent agent\_id to CV  Requires JSON body:  ***{***  ***“ip” : ipaddr,***  ***“keyfile\_data” : base64,***  ***“keyfile\_name” : string,*** (*opt*)  ***“file\_data” : base64,***  ***“file\_name” : string,*** (*opt*)  ***“ca\_dir” : string,***  ***“ca\_dir\_pw” : string,***  ***“include\_dir\_data” : base64,***  ***“include\_dir\_name” : string,***  ***}*** | Terminate agent agent\_id |
| /v2/logs | Get terminal log data |  |  |  |
| /v2/logs/{logType:string}    ex.)  */v2/logs/tenant?pos=#* | Get terminal log data for given logType  Optional query parameters:  ***pos : int,*** (*opt*) |  |  |  |

**RESTful API Responses for Keylime (*v2*)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *tenant -> cloud verifier (CV)* | GET (retrieve) | PUT (replace/edit) | POST (new) | DELETE (remove) |
| /v2/agents/ | “results” = {  “uuids” = [*uuid1*, *uuid2*, …]  } |  |  |  |
| /v2/agents/*{agent\_id:UUID}* | “results” = {  “operational\_state”,  “v”,  “ip”,  “port”,  “tpm\_policy”,  “vtpm\_policy”,  “metadata”,  “ima\_whitelist”,  “accept\_tpm\_hash\_algs”,  “accept\_tpm\_encryption\_algs”,  “accept\_tpm\_signing\_algs”,  “hash\_alg”,  “enc\_alg”,  “sign\_alg”,  } |  | “results” = {} | “results” = {} |
| /v2/agents/*{agent\_id:UUID}*/reactivate |  | “results” = {} |  |  |
| /v2/agents/*{agent\_id:UUID}*/stop |  | “results” = {} |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *verifier/tenant -> agent* | GET (retrieve) | PUT (replace/edit) | POST (new) | DELETE (remove) |
| /v2/keys/ |  |  |  |  |
| /v2/keys/pubkey | “results” = { “pubkey” } |  |  |  |
| /v2/keys/vkey |  |  | “results” = {} |  |
| /v2/keys/ukey |  |  | “results” = {} |  |
| /v2/keys/verify  ex.)  */v2/keys/verify?challenge=#* | “results” = { “hmac” } |  |  |  |
| /v2/quotes/ |  |  |  |  |
| /v2/quotes/integrity  ex.)  */v2/quotes/integrity?nonce=#&mask=#&vmask=#&partial=#* | “results” = {  “quote”,  “tpm\_version”,  “hash\_alg”,  “enc\_alg”,  “sign\_alg”,  “pubkey”, (*if not partial*)  } |  |  |  |
| /v2/quotes/identity  ex.)  */v2/quotes/identity?nonce=#* | “results” = {  “quote”,  “tpm\_version”,  “hash\_alg”,  “enc\_alg”,  “sign\_alg”,  “pubkey”,  } |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *verifier/agent/tenant -> registrar* | GET (retrieve) | PUT (replace/edit) | POST (new) | DELETE (remove) |
| /v2/agents/ | “results” = {  “uuids” = [*uuid1*, *uuid2*, …]  } |  |  |  |
| /v2/agents/{agent\_id:UUID} | “results” = {  “aik”,  “ek”,  “ekcert”,  “regcount”,  “provider\_keys”, (*ONLY IF VIRTUAL*)  } |  | “results” = {  “blob” = *EncEK(AIK)*,  } | “results” = {} |
| /v2/agents/{agent\_id:UUID}/activate |  | “results” = {} |  |  |
| /v2/agents/{agent\_id:UUID}/vactivate |  | “results” = {} |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Tenant webapp* | GET (retrieve) | PUT (replace/edit) | POST (new) | DELETE (remove) |
| /v2/agents/ | “results” = {  “uuids” = [*uuid1*, *uuid2*, …]  } |  |  |  |
| /v2/agents/{agent\_id:UUID} | “results” = {  “id”,  “operational\_state”,  “v”,  “ip”,  “port”,  “tpm\_policy”,  “vtpm\_policy”,  “metadata”,  “ima\_whitelist”,  } | “results” = {} | “results” = {} | “results” = {} |
| /v2/logs |  |  |  |  |
| /v2/logs/{logType:string}    ex.)  */v2/logs/tenant?pos=#* | “results” = {  “log” = [*string*, *string*, …]  *(lines)*  } |  |  |  |

**JSON Response Format (v2)**

|  |
| --- |
| {  “code” : int,  “status” : string,  “results” : JSON,  } |

**HTTP Status Codes**

|  |  |  |
| --- | --- | --- |
| *Code (Number)* | Code (Textual) | Description |
| 200 | Success (OK) | Results return as requested |
| 201 | Created | A resource was created as requested |
| 400 | Bad request | Your request was incorrect / invalid / malformed |
| 403 | Forbidden | You do not have the required permissions |
| 404 | Not found | API request error |
| 405 | Method not allowed | Requested method not allowed for this resource (provides ‘Allow’ header) |
| 409 | Conflict | Request could not be completed due to resource state conflict |
| 500 | Internal error | Internal problem (e.g., server, database) |
| 501 | Not implemented | Request method not yet supported for this resource |