



## Local/Offline Mode Commands

[hub.offline.credentials.updated](#)

[Request](#)

[Response](#)

[Example](#)

[hub.offline.login.ui](#)

[Request](#)

[Response](#)

[Example](#)

[hub.offline.login.hub](#)

[Request](#)

[Response](#)

[Example](#)

[hub.offline.anonymous\\_access.set](#)

[Request](#)

[Response](#)

[Example](#)

[hub.offline.insecure\\_access.enabled.set](#)

[Request](#)

[Response](#)

[Example](#)

[cloud.access\\_keys\\_controller\\_sync](#)

[Request](#)

[Response](#)

[Example](#)

### hub.offline.credentials.updated

**Connection type:**

CLOUD

Notify controller about new local access data availability. The controller should request this data soon via [cloud.access\\_keys\\_controller\\_sync](#) request if it still has an outdated version.

**Request:**

Field	Type	Required	Description
newVersion	int	+	Version of new local access data

**Response:**

empty result

**Example:****request:**

```
{
  "method": "hub.offline.credentials.updated",
  "id": "_ID_",
  "params": {
    "newVersion": 45
  }
}
```

**response:**

```
{
  "error": null,
  "id": "_ID_",
  "result": {}
}
```

## hub.offline.login.ui

**Connection type:**

LOCAL

**Permissions:**

EZLO:READ

Login to Controller via direct connection as user

#### Request:

Field	Type	Required	Description
user	string	+	User identifier (UUID)
token	string	+	Access token

#### Response:

empty result

#### Errors:

Code	Message	Data
-32600	Bad request,	rpc.params.notfound.
-32602	Wrong params	rpc.params.invalid
-32500	Bad password	user.login.badpassword
-32500	Already logged in	user.login.alreadylogged

#### Example:

##### request:

```
{
  "method": "hub.offline.login.ui",
  "id": "_ID_",
  "params": {
    "User": "user.identifier",
    "token": "Hs87Kns41106743NMS2"
  }
}
```

**response:**

```
{
  "error": null,
  "id": "-ID-",
  "result": {}
}
```

## hub.offline.login.hub

**Connection type:**

LOCAL

Login to Controller via direct connection as controller

**Request:**

Field	Type	Required	Description
hub	string	+	Controller identifier (UUID)
token	string	+	Access token

**Response:**

empty result

**Errors:**

Code	Message	Data
-32600	Bad request,	rpc.params.notfound.
-32602	Wrong params	rpc.params.invalid
-32500	Bad password	user.login.badpassword

-32500	Already logged in	user.login.alreadylogged
--------	-------------------	--------------------------

### Example:

#### request:

```
{
  "method": "hub.offline.login.hub",
  "id": "_ID_",
  "params": {
    "hub": "39c681f9-c3e4-44b2-8e77-b8c22c305b41",
    "token": "Hs87Kns41106743NMS2"
  }
}
```

#### response:

```
{
  "error": null,
  "id": "_ID_",
  "result": {}
}
```

## hub.offline.anonymous\_access.set

### Connection type:

ANY
-----

Enable or disable anonymous access (without authentication via [hub.offline.login.ui](#) for requests via WS API or HTTP basic authentication for requests via HTTP API). Current status can be found using [hub.info.get](#) request.

### Request:

Field	Type	Required	Description
enabled	bool	+	Enable or disable anonymous access

#### Response:

```
empty result
```

#### Errors:

Code	Message	Data
-32600	Bad request,	rpc.params.notfound.

#### Example:

##### request:

```
{
  "method": "hub.offline.anonymous_access.enabled.set",
  "id": "_ID_",
  "params": {
    "Enabled": true
  }
}
```

##### response:

```
{
  "error": null,
  "id": "_ID_",
  "result": {}
}
```

[hub.offline.insecure\\_access.enabled.set](#)

**Connection type:**

ANY

Enable or disable insecure access (WS and HTTP as opposed to secure WSS and HTTPS). You may want to enable it if self-signed certificates used on the controller are not supported in your client software. Current status can be found using [hub.info.get](#) request.

**Request:**

Field	Type	Required	Description
enabled	bool	+	Enable or disable insecure access

**Response:**

empty result

**Errors:**

Code	Message	Data
-32600	Bad request,	rpc.params.notfound.

**Example:****request:**

```
{
  "method": "hub.offline.insecure_access.enabled.set",
  "id": "_ID_",
  "params": {
    "Enabled": true
  }
}
```

**response:**

```
{
  "error": null,
  "id": "_ID_",
  "result": {}
}
```

## cloud.access\_keys\_controller\_sync

### Hub type:

CLOUD

Title: ask cloud for new local access data

Description: Controller will ask for local access data when it have no (first start, after reset), or after cloud notified controller about new local access data with [hub.offline.credentials.updated](#) command.

### Request:

Field	Type	Required	Description
<b>version</b>	int	-	Current local access data version on controller (omitted if controller have no local access data)

### Response:

Field	Type	Required	Description
<b>clients</b>	object	+	Credentials for users/controllers/etc that connect to current controller



<b>clients.{UUID}</b>	array of strings	-	Array of SHA256 hashed keys (access tokens) for user specified by UUID
<b>peers</b>	object	+	Credentials for controllers or devices you can both connect to and receive connections from
<b>peers.{UUID}</b>	object	+	Controller info
<b>peers.{UUID}.id</b>	string	+	Controller serial
<b>peers.{UUID}.tokens</b>	array of strings	+	Array of plain text keys (access tokens) for peer specified by UUID
<b>current</b>	array	+	Data for the controller doing the sync API call
<b>current[].certificate</b>	string	+	Certificate in PEM format for secure websocket connection
<b>current[].private</b>	string	+	Private key in PEM format for secure websocket connection
<b>current[].public_parent</b>	string	+	SSL Certificates Chain file to verify all controller's certificates (during controller-to-controller connection controller-client will use it to verify controller-server's certificate)

### Example:

#### request:

```
{
  "id": "_ID_",
  "api": "3",
  "method": "cloud.access_keys_controller_sync",
  "params": {
    "Version": 1
  }
}
```

#### response:

```
{
  "error": null,,
  "method": "cloud.access_keys_controller_sync",
  "id": "_ID_",
  "result": {
    "clients": 1
    "95d8cd00-5bba-46e5-ae33-f1e4b2d65e6e": [
      "eb7a599e4a991e61809ff9f0883e5801cb33e96ea165350b90a1a210023596e7",
      "1b191ccc8ca77590a28a48067f875047556a890acd29ec50fb046d718175bec7"
    ],
    "708e3376-5cb1-4fa8-923f-cda75b9e9e33": [
      "bf7d0763877fe1d39308176e991a187b6f8c25d2c5617ea6d2b71e03400eae6e"
    ]
  },
  "peers": {
    "c89723f7-03a3-423d-a556-7eba4270d064" : {
      "id": "70010111",
      "tokens": [
        "c89723f7-03a3-423d-a556-7eba4270d064-key-12-02",
        "c89723f7-03a3-423d-a556-7eba4270d064-key-01-03"
      ]
    }
  },
  "current": [
    {
      "certificate": "-----BEGIN CERTIFICATE-----\nMIIE...\n-----END CERTIFICATE-----",
      "private": "-----BEGIN PRIVATE KEY-----\nMIIE...\n-----END PRIVATE KEY-----",
      "public_parent": "-----BEGIN CERTIFICATE-----\nMIIE...\n-----END CERTIFICATE-----"
    },
    {
      "certificate": "-----BEGIN CERTIFICATE-----\nMIIE...\n-----END CERTIFICATE-----",
      "private": "-----BEGIN PRIVATE KEY-----\nMIIE...\n-----END PRIVATE KEY-----",
      "public_parent": "-----BEGIN CERTIFICATE-----\nMIIE...\n-----END CERTIFICATE-----"
    }
  ],
  "version": 12
}
}
```