

Exosense Server JSON-RPC Reference Manual

Copyright 2012, 2013 Feuerlabs, Inc. All rights reserved.

2013-02-04

Contents

1	Module: exodm	13
2	exodm:create-user	14
2.1	Request	14
2.1.1	Elements	14
2.2	Reply	14
2.2.1	Elements	15
3	exodm:delete-user	16
3.1	Request	16
3.1.1	Elements	16
3.2	Reply	16
3.2.1	Elements	16
4	exodm:create-account	17
4.1	Request	17
4.1.1	Elements	17
4.2	Reply	17
4.2.1	Elements	17
5	exodm:update-account	18
5.1	Request	18
5.1.1	Elements	18
5.2	Reply	18
5.2.1	Elements	18
6	exodm:delete-account	19
6.1	Request	19
6.1.1	Elements	19
6.2	Reply	19
6.2.1	Elements	19

7	exodm:list-accounts	20
7.1	Request	20
7.1.1	Elements	20
7.2	Reply	20
7.2.1	Elements	21
8	exodm:add-user-to-account	22
8.1	Request	22
8.1.1	Elements	22
8.2	Reply	22
8.2.1	Elements	22
9	exodm:remove-account-users	23
9.1	Request	23
9.1.1	Elements	23
9.2	Reply	23
9.2.1	Elements	23
10	exodm:list-account-users	24
10.1	Request	24
10.1.1	Elements	24
10.2	Reply	24
10.2.1	Elements	25
11	exodm:add-user-to-role	26
11.1	Request	26
11.1.1	Elements	26
11.2	Reply	27
11.2.1	Elements	27

12 exodm:remove-user-from-role	28
12.1 Request	28
12.1.1 Elements	28
12.2 Reply	29
12.2.1 Elements	29
13 exodm:add-role-execution-permission	30
13.1 Request	30
13.1.1 Elements	30
13.2 Reply	31
13.2.1 Elements	31
14 exodm:remove-role-execution-permission	32
14.1 Request	32
14.1.1 Elements	32
14.2 Reply	33
14.2.1 Elements	33
15 exodm:list-rpc-roles	34
15.1 Request	34
15.1.1 Elements	34
15.2 Reply	35
15.2.1 Elements	35
16 exodm:list-account-roles	36
16.1 Request	36
16.1.1 Elements	36
16.2 Reply	36
16.2.1 Elements	37

17 exodm:create-device	38
17.1 Request	38
17.1.1 Elements	39
17.2 Reply	39
17.2.1 Elements	39
18 exodm:update-device	40
18.1 Request	40
18.1.1 Elements	40
18.2 Reply	40
18.2.1 Elements	41
19 exodm:lookup-device	42
19.1 Request	42
19.1.1 Elements	42
19.2 Reply	42
19.2.1 Elements	43
20 exodm:list-devices	44
20.1 Request	44
20.1.1 Elements	44
20.2 Reply	44
20.2.1 Elements	45
21 exodm:delete-devices	46
21.1 Request	46
21.1.1 Elements	46
21.2 Reply	46
21.2.1 Elements	46

22 exodm:create-device-group	47
22.1 Request	47
22.1.1 Elements	47
22.2 Reply	48
22.2.1 Elements	48
23 exodm:delete-device-group	49
23.1 Request	49
23.1.1 Elements	49
23.2 Reply	49
23.2.1 Elements	49
24 exodm:list-device-groups	50
24.1 Request	50
24.1.1 Elements	50
24.2 Reply	50
24.2.1 Elements	51
25 exodm:list-device-group-members	52
25.1 Request	52
25.1.1 Elements	52
25.2 Reply	52
25.2.1 Elements	53
26 exodm:add-device-group-members	54
26.1 Request	54
26.1.1 Elements	55
26.2 Reply	55
26.2.1 Elements	55

27 exodm:remove-device-group-members	56
27.1 Request	56
27.1.1 Elements	56
27.2 Reply	56
27.2.1 Elements	57
28 exodm:update-device-group	58
28.1 Request	58
28.1.1 Elements	58
28.2 Reply	58
28.2.1 Elements	59
29 exodm:create-device-type	60
29.1 Request	60
29.1.1 Elements	60
29.2 Reply	61
29.2.1 Elements	61
30 exodm:update-device-type	62
30.1 Request	62
30.1.1 Elements	62
30.2 Reply	63
30.2.1 Elements	63
31 exodm:delete-device-type	64
31.1 Request	64
31.1.1 Elements	64
31.2 Reply	64
31.2.1 Elements	64

32 exodm:list-device-types	65
32.1 Request	65
32.1.1 Elements	65
32.2 Reply	65
32.2.1 Elements	66
33 exodm:list-device-type-devices	67
33.1 Request	67
33.1.1 Elements	67
33.2 Reply	67
34 exodm:create-config-set	69
34.1 Request	69
34.1.1 Elements	69
34.2 Reply	70
34.2.1 Elements	70
35 exodm:update-config-set	71
35.1 Request	71
35.1.1 Elements	71
35.2 Reply	71
35.2.1 Elements	72
36 exodm:delete-config-set	73
36.1 Request	73
36.1.1 Elements	73
36.2 Reply	73
36.2.1 Elements	74
37 exodm:list-config-sets	75
37.1 Request	75
37.1.1 Elements	75
37.2 Reply	75
37.2.1 Elements	76

38 exodm:list-config-set-members	77
38.1 Request	77
38.1.1 Elements	77
38.2 Reply	77
38.2.1 Elements	78
39 exodm:add-config-set-members	79
39.1 Request	79
39.1.1 Elements	79
39.2 Reply	80
39.2.1 Elements	80
40 exodm:remove-config-set-members	81
40.1 Request	81
40.1.1 Elements	81
40.2 Reply	81
40.2.1 Elements	82
41 exodm:push-config-set	83
41.1 Request	83
41.1.1 Elements	83
41.2 Reply	83
41.2.1 Elements	84
41.3 Notification: exodm:push-config-set-callback	84
41.3.1 Elements	84
42 exodm:create-yang-module	85
42.1 Request	85
42.1.1 Elements	85
42.2 Reply	86
42.2.1 Elements	86

43 exodm:delete-yang-module	87
43.1 Request	87
43.1.1 Elements	87
43.2 Reply	87
43.2.1 Elements	88
44 exodm:list-yang-modules	89
44.1 Request	89
44.1.1 Elements	89
44.2 Reply	89
44.2.1 Elements	90
45 exodm:create-package	91
45.1 Request	91
45.1.1 Elements	92
45.2 Reply	93
45.2.1 Elements	93
46 exodm:add-package-to-device-type	94
46.1 Request	94
46.1.1 Elements	94
46.2 Reply	95
46.2.1 Elements	95
47 exodm:remove-package-from-device-type	96
47.1 Request	96
47.1.1 Elements	96
47.2 Reply	97
47.2.1 Elements	97

48 exodm:install-package	98
48.1 Request	98
48.1.1 Elements	99
48.2 Reply	99
48.2.1 Elements	100
49 exodm:upgrade-package	101
49.1 Request	101
49.1.1 Elements	102
49.2 Reply	102
49.2.1 Elements	102
50 exodm:uninstall-package	103
50.1 Request	103
50.1.1 Elements	103
50.2 Reply	104
50.2.1 Elements	104
50.3 Notification: exodm:package-operation-notification	104
50.3.1 Elements	105
51 exodm:list-packages	106
51.1 Request	106
51.1.1 Elements	106
51.2 Reply	106
51.2.1 Elements	107
52 exodm:list-device-types-with-package	108
52.1 Request	108
52.1.1 Elements	108
52.2 Reply	108
52.2.1 Elements	109

53 exodm:list-packages-on-device-type	110
53.1 Request	110
53.1.1 Elements	110
53.2 Reply	110
53.2.1 Elements	111
54 exodm:list-devices-with-package	112
54.1 Request	112
54.1.1 Elements	112
54.2 Reply	113
54.2.1 Elements	113
55 exodm:update-package-config	114
55.1 Request	114
55.1.1 Elements	114
55.2 Reply	115

1 Module: exodm

2 exodm:create-user

2.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:create-user",
  "id": "",
  "params": {
    "uname": "",
    "email": "",
    "password": "",
    "fullname": "",
    "phone": "",
    "skype": "",
    "alias": [
      "alias": ""
    ]
  }
}
```

Create a new user

This command creates a new user that can be given access to accounts and assigned roles. The authorizing user must have either master or admin roles assigned to it for this command to be executed. The created user can be granted access to accounts through roles using the [exodm:add-user-to-role](#) command.

2.1.1 Elements

alias (**type:** string; [mandatory: false])
email (**type:** string; [mandatory: true])
fullname (**type:** string; [mandatory: false])
password (**type:** string; [mandatory: true])
phone (**type:** string; [mandatory: false])
skype (**type:** string; [mandatory: false])
uname (**type:** string; [mandatory: true])

2.2 Reply

```
{
```

```
    "jsonrpc": "2.0",  
    "id": "",  
    "result": {  
        "result": ""  
    }  
}
```

2.2.1 Elements

result Result of the operation (**type:** enumeration; [mandatory: false])

3 exodm:delete-user

3.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:delete-user",
  "id": "",
  "params": {
    "uname": ""
  }
}
```

Delete an existing user

This command deletes an existing user previously created through an **exodm:create-user** command. The user will be removed from all accounts and roles it was assigned to prior to being deleted.

3.1.1 Elements

uname The name of the user to delete. (**type:** string; [mandatory: true])

3.2 Reply

```
{
  "jsonrpc": "2.0",
  "id": "",
  "result": {
    "result": ""
  }
}
```

3.2.1 Elements

result Result of the operation (**type:** enumeration; [mandatory: false])

4 exodm:create-account

4.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:create-account",
  "id": "",
  "params": {
    "name": ""
  }
}
```

Create a new account

This call, which must be invoked by the sysadm user, creates a new Exosense Account. The created account can give access permission to users that will manage the devices, device groups, packages, etc, owned by the account.

When the account is created, an account admin user, [name]-admin will be also be automatically created. This user acts as the initial administrator for the account and can be used to manage all resources owned by the account. The name of the account admin is the value of the name argument with an -admin suffix.

4.1.1 Elements

name The name of the account. This name has to be unique across all accounts created in the Exosense Server. (**type:** string; [mandatory: true])

4.2 Reply

```
{
  "jsonrpc": "2.0",
  "id": "",
  "result": {
    "result": ""
  }
}
```

4.2.1 Elements

result The result of the operation. (**type:** enumeration; [mandatory: false])

5 exodm:update-account

5.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:update-account",
  "id": "",
  "params": {
    "name": ""
  }
}
```

Update an existing account

This call modifies the name of an account previously created with a **exodm:create-account** command. The new name of the account must be unique across all accounts in the Exosense Server.

5.1.1 Elements

name The name of the account. This name has to be unique across all accounts created in the Exosense Server. (**type:** string; [mandatory: false])

5.2 Reply

```
{
  "jsonrpc": "2.0",
  "id": "",
  "result": {
    "result": ""
  }
}
```

5.2.1 Elements

result The result of the operation. (**type:** enumeration; [mandatory: false])

6 exodm:delete-account

6.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:delete-account",
  "id": "",
  "params": {
    "name": ""
  }
}
```

****Delete an existing account**

****This command will remove the account specified by the name argument. If the given account owns any devices, device types, device groups, packages, configuration sets or yang specifications, they will be deleted together with the account.**

6.1.1 Elements

name The name of the account to delete. (**type:** string; [mandatory: true])

6.2 Reply

```
{
  "jsonrpc": "2.0",
  "id": "",
  "result": {
    "result": ""
  }
}
```

6.2.1 Elements

result The result of the operation. (**type:** enumeration; [mandatory: false])

7 exodm:list-accounts

7.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:list-accounts",
  "id": "",
  "params": {
    "n": "",
    "previous": ""
  }
}
```

List all accounts in the Exosense Server

This command will list all accounts created in the Exosense Server. This command can only be executed by the sysadm user.

This command allows for chunks of the result set to be returned to facilitate a sequential retrieval of accounts.

7.1.1 Elements

n Maximum number of entries to return. More accounts may be available after the last returned element. (**type:** uint32; [mandatory: true])

previous Start the fetch at the first account after the account with the name specified by this argument. Use "" if the fetch is to start from the beginning (**type:** string; [mandatory: true])

7.2 Reply

```
{
  "jsonrpc": "2.0",
  "id": "",
  "result": {
    "accounts": [
      "accounts": ""
    ]
  }
}
```

7.2.1 Elements

accounts List of account names (**type:** string; [mandatory: false])

8 exodm:add-user-to-account

8.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:add-user-to-account",
  "id": "",
  "params": {
    "account": "",
    "uname": ""
  }
}
```

Add a user to an account

This command adds the specified user to the given account. Once a user has been added to an account, the user can be assigned to roles in the given account through the **exodm:add-user-to-role** command. The user will then be able to execute all RPCs, assigned to the role through the **exodm:add-role-execution-permission** command, on all devices owned by the account.

8.1.1 Elements

account The account to add the user to (**type:** string; [mandatory: false])

uname The user to add to the account. (**type:** string; [mandatory: false])

8.2 Reply

```
{
  "jsonrpc": "2.0",
  "id": "",
  "result": {
    "result": ""
  }
}
```

8.2.1 Elements

result The result of the operation. (**type:** enumeration; [mandatory: false])

9 exodm:remove-account-users

9.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:remove-account-users",
  "id": "",
  "params": {
    "account": "",
    "uname": ""
  }
}
```

Remove a user from an account

This command removes the specified user to the given account. Once a user has been removed to an account, any previous RPCs that the user could execute on devices owned by the account will become inaccessible.

9.1.1 Elements

account The account to remove the user from (**type:** string; [mandatory: false])

uname The user to remove from the account. (**type:** string; [mandatory: false])

9.2 Reply

```
{
  "jsonrpc": "2.0",
  "id": "",
  "result": {
    "result": ""
  }
}
```

9.2.1 Elements

result The result of the operation. (**type:** enumeration; [mandatory: false])

10 exodm:list-account-users

10.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:list-account-users",
  "id": "",
  "params": {
    "account": "",
    "n": "",
    "previous": ""
  }
}
```

****List all users with access to the specified account**

****This command will list all users who have been given access to the given account through `exodm:add-user-to-account` commands.**

This command allows for chunks of the result set to be returned to facilitate a sequential retrieval of user names.

10.1.1 Elements

account The account, accessible by the authorizing user, to list the users for. If no account is given, the authorizing user assumes to have access to only one account, which will be used. If no account is given, and the user has access to multiple accounts, an error is returned. (**type:** string; [mandatory: true])

n Maximum number of entries to return. More users may be available after the last returned element. (**type:** uint32; [mandatory: true])

previous Start the fetch at the first user after the user with the name specified by this argument. Use "" if the fetch is to start from the beginning (**type:** string; [mandatory: true])

10.2 Reply

```
{
  "jsonrpc": "2.0",
  "id": "",
  "result": {
    "users": [
      "users": ""
    ]
  }
}
```



```
    ]  
  }  
}
```

10.2.1 Elements

users The (partial) list of user names with access to the given account. (**type:** string; [mandatory: false])

11 exodm:add-user-to-role

11.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:add-user-to-role",
  "id": "",
  "params": {
    "account": "",
    "role": "",
    "names": [
      "names": ""
    ]
  }
}
```

Add user(s) to a role in an account

This command gives one or more users access to the resources of an account by adding them to a role. Each role will give access to a predefined set of core Exosense Server JSON-RPC commands, as listed in the “Exosense JSON-RPC command Role assignment” chapter of the “Exosense Server Operations Manual”. In addition to this, roles can also be given additional execution rights to package RPCs through the `exodm:add-role-execution-permission` command. See the “Package” chapter of the “Exosense Operations Manual” and `exodm:create-package` for details on RPCs and their execution rights.

Once one or more roles have been assigned to a user, that user can execute all JSON-RPC commands that those roles have execution rights for on the given account.

The `exodm:list-account-roles` command can be used to retrieve a list of all available roles for an account that a user can be assigned to.

11.1.1 Elements

account The account, accessible by the authorizing user, to which the role belongs. If no account is given, the authorizing user assumes to have access to only one account, which will be used. If no account is given, and the user has access to multiple accounts, an error is returned. (**type:** string; [mandatory: true])

names Users to add to account. The users, previously created through the `exodm:create-user` or `exodm:create-account` commands, to be assigned the given role for this account. (**type:** string; [mandatory: false])

role The role to assign to the user for the specified account. The value of this argument must be from the list of roles returned by [exodm:list-account-roles](#) (**type:** string; [mandatory: true])

11.2 Reply

```
{
  "jsonrpc": "2.0",
  "id": "",
  "result": {
    "result": ""
  }
}
```

11.2.1 Elements

result Result of the operation (**type:** enumeration; [mandatory: false])

12 exodm:remove-user-from-role

12.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:remove-user-from-role",
  "id": "",
  "params": {
    "account": "",
    "role": "",
    "unames": [
      "unames": ""
    ]
  }
}
```

Remove user(s) from a role

This command retracts one or more user's access to the resources of an account by removing the user from a role. The user have previously been added to the role through a [exodm:add-user-to-role](#) command. Once the user has been removed from the role, it can no longer run the JSON-RPC commands that the role had execution rights for (unless the user is also assigned other roles that have execution rights for the same commands). See the “Package” chapter of the “Exosense Operations Manual” and [exodm:create-package](#) for details on RPCs and Yang specifications.

12.1.1 Elements

account The account, accessible by the authorizing user, to which the role belongs. If no account is given, the authorizing user assumes to have access to only one account, which will be used. If no account is given, and the user has access to multiple accounts, an error is returned. (**type:** string; [mandatory: true])

role The role to strip the given user of for the specified account. The value of this argument must be from the list of roles returned by [exodm:list-account-roles](#) (**type:** string; [mandatory: true])

unames Users to strip of the specified role for this account. Listed users must previously have been assigned the role through an [exodm:add-user-to-role](#) command. (**type:** string; [mandatory: false])

12.2 Reply

```
{
  "jsonrpc": "2.0",
  "id": "",
  "result": {
    "result": ""
  }
}
```

12.2.1 Elements

result Result of the operation (**type:** enumeration; [mandatory: false])

13 exodm:add-role-execution-permission

13.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:add-role-execution-permission",
  "id": "",
  "params": {
    "account": "",
    "role": "",
    "package-name": "",
    "rpc-name": ""
  }
}
```

Add permission execution for a role to an RPC

This command adds execution rights for a role to an RPC in a package owned by the given account. Once this command has executed, all users assigned to the role will be able to execute the given RPC on all devices, owned by the account, that the associated package has been installed on.

Users can be added to the role through the [exodm:add-user-to-role](#) command.

If multiple roles have execution rights to a single RPC within a package, any user who is assigned to one or more of these roles can execute the RPC. Conversely, a user must be removed from all roles with execution rights to an RPC in order to revoke the execution right for that user.

13.1.1 Elements

account The account, accessible by the authorizing user, to which the role and package belongs. If no account is given, the authorizing user assumes to have access to only one account, which will be used. If no account is given, and the user has access to multiple accounts, an error is returned. (**type:** string; [mandatory: false])

package-name Name of package to which to add role execution rights. The package has previously been created through a [exodm:create-package](#) command (**type:** string; [mandatory: true])

role Name of role to add execution permission to. (**type:** string; [mandatory: true])

rpc-name Name of the RPC in the package to add role execution rights to. The name of the RPC should include the module prefix and a colon (fleetmanager:get-position) (**type:** string; [mandatory: true])

13.2 Reply

```
{
  "jsonrpc": "2.0",
  "id": "",
  "result": {
    "result": ""
  }
}
```

13.2.1 Elements

result (**type:** enumeration; [mandatory: false])

14 exodm:remove-role-execution-permission

14.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:remove-role-execution-permission",
  "id": "",
  "params": {
    "account": "",
    "role": "",
    "package-name": "",
    "rpc-name": ""
  }
}
```

Remove permission for a role to execute an RPC

This command removes execution rights for a role from an RPC in a package owned by the given account. Once this command has executed, all users assigned to the role will no longer have permission to execute the given RPC on devices, owned by the account, that the associated package has been installed on.

If multiple roles have execution rights to a single RPC within a package, any user who is assigned to one or more of these roles can execute the RPC. Conversely, a user must be removed from all roles with execution rights to an RPC in order to revoke the execution right for that user.

14.1.1 Elements

account The account, accessible by the authorizing user, to which the role and package belongs. If no account is given, the authorizing user assumes to have access to only one account, which will be used. If no account is given, and the user has access to multiple accounts, an error is returned. (**type:** string; [mandatory: false])

package-name Name of package fromn which to remove role execution rights. (**type:** string; [mandatory: true])

role Name of role to remove execution permission from. (**type:** string; [mandatory: true])

rpc-name Name of the RPC in the package to remove role execution rights from. The name of the RPC should include the module prefix and a colon (fleetmanager:get-position) (**type:** string; [mandatory: true])

14.2 Reply

```
{
  "jsonrpc": "2.0",
  "id": "",
  "result": {
    "result": ""
  }
}
```

14.2.1 Elements

result (**type:** enumeration; [mandatory: false])

15 exodm:list-rpc-roles

15.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:list-rpc-roles",
  "id": "",
  "params": {
    "account": "",
    "package-name": "",
    "rpc-name": "",
    "n": "",
    "previous": ""
  }
}
```

List all roles that have exeuction right for an RPC

This command will list all roles with execution rights for the given account/package/RPC combination

This command allows for chunks of the result set to be returned to facilitate a sequential retrieval of roles.

15.1.1 Elements

account The account, accessible by the authorizing user, to which the role and package belongs.If no account is given, the authorizing user assumes to have access to only one account, which will be used. If no account is given, and the user has access to multiple accounts, an error is returned. (**type:** string; [mandatory: false])

n Maximum number of entries to return. More roles may be available after the last returned element. (**type:** uint32; [mandatory: false])

package-name Name of package to list role access for, (**type:** string; [mandatory: true])

previous Start the fetch at the first role after the role with the name specified by this argument. Use "" if the fetch is to start from the beginning (**type:** uint32; [mandatory: false])

rpc-name Name of RPC inside the package to list role access for. The name of the RPC should include the module prefix and a colon (fleetmanager:get-position). (**type:** string; [mandatory: true])

15.2 Reply

```
{
  "jsonrpc": "2.0",
  "id": "",
  "result": {
    "roles": [
      "roles": ""
    ]
  }
}
```

15.2.1 Elements

roles List of roles with execution right to the given RPC in the specified package owned by the account. (**type:** string; [mandatory: false])

16 exodm:list-account-roles

16.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:list-account-roles",
  "id": "",
  "params": {
    "account": "",
    "n": "",
    "previous": ""
  }
}
```

List all roles defined for account

This command will return all roles in account that can be assigned to users in order for them to access resources owned by the account.

This command allows for chunks of the result set to be returned to facilitate a sequential retrieval of roles.

16.1.1 Elements

account The account, accessible by the authorizing user, to which the roles belongs. If no account is given, the authorizing user assumes to have access to only one account, which will be used. If no account is given, and the user has access to multiple accounts, an error is returned. (**type:** string; [mandatory: true])

n Maximum number of entries to return. More roles may be available after the last returned element. (**type:** uint32; [mandatory: true])

previous Start the fetch at the first role after the role with the name specified by this argument. Use "" if the fetch is to start from the beginning (**type:** string; [mandatory: true])

16.2 Reply

```
{
  "jsonrpc": "2.0",
  "id": "",
  "result": {
    "roles": [
```

```

        "roles": ""
    ]
}

```

16.2.1 Elements

roles List of role names assignable to users. (**type:** string; [mandatory: false])

17 exodm:create-device

17.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:create-device",
  "id": "",
  "params": {
    "account": "",
    "device-id": "",
    "device-type": "",
    "attributes": ""
  }
}
```

Create a new device in the Exosense Server

This command sets up a new device to be owned and managed by the given account. The mandatory device-id must be unique across all devices and device groups under the owning account. The device type is previously created through a [exodm:create-device-type](#) command.

The new device can optionally have attributes set for it, where the structure of these attributes is defined by the provided device type. The device type has a yang module, created through an [exodm:create-yang-module](#) command, tied to it. Inside this module, a container defines the attribute structure that should be applied to all devices assigned to the given device type.

Unlink device configuration data (see [exodm:create-package](#)), device attributes will not be sent to the device. Instead they will be stored locally on the Exosense Server.

The attributes can be set through this command and [exodm:update-device](#), and retrieved through [exodm:lookup-device](#). Protocol plugins can also access the device attributes to locate network addresses and other information necessary for the protocol to communicate with a device.

Once a device has been provisioned through this command, the protocol plugin associated with the assigned device type can match an incoming device ID from the connected device with device attribute data in order to retrieve encryption keys, passwords, network addresses, certificates, etc, that can be used by the plugin to authenticate the device and secure communication with it. Please note that all device attributes, apart from the mandatory device type and device-id, are opaque to the Exosense Server. It is up to the yang-file and protocol plugin author to agree on the structure of the attributes assigned to a device of the given device type.

17.1.1 Elements

account The account, accessible by the authorizing user, to which the device type belongs. If no account is given, the authorizing user assumes to have access to only one account, which will be used. If no account is given, and the user has access to multiple accounts, an error is returned. (**type:** string; [mandatory: false])

attributes Attributes to set for this device. The structure of the attributes is defined by the yang module provided to the **exodm:create-device-type** command that created the device type that the device is assigned to. (**type:** XML; [mandatory: false])

device-id Account-wide unique device-id (**type:** string; [mandatory: true])

device-type Type of device, as given by a previous **exodm:create-device-type** call. (**type:** string; [mandatory: true])

17.2 Reply

```
{
  "jsonrpc": "2.0",
  "id": "",
  "result": {
    "result": ""
  }
}
```

17.2.1 Elements

result Result of the operation (**type:** enumeration; [mandatory: false])

18 exodm:update-device

18.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:update-device",
  "id": "",
  "params": {
    "account": "",
    "attributes": ""
  }
}
```

Update the attributes of an existing device

This command updates the attributes of the given device. Please note that the device-id and device type of an existing device cannot be modified. For these operations, the device needs to be deleted with [exodm:delete-device] and then re-created with [exodm:create-device](#).

The attributes that can be updated for a device is defined by the assigned device type. Please see [exodm:create-device](#) for information on how device attributes are specified and managed.

18.1.1 Elements

account The account, accessible by the authorizing user, to which the device type belongs. If no account is given, the authorizing user assumes to have access to only one account, which will be used. If no account is given, and the user has access to multiple accounts, an error is returned. (**type:** string; [mandatory: false])

attributes Attributes to set for the device. The structure of the attributes is defined by the yang module provided to the [exodm:create-device-type](#) command that created the device type that the device is assigned to. (**type:** XML; [mandatory: false])

18.2 Reply

```
{
  "jsonrpc": "2.0",
  "id": "",
  "result": {
    "result": ""
  }
}
```



```
    }  
}
```

18.2.1 Elements

result Result of the operation (**type:** enumeration; [mandatory: false])

19 exodm:lookup-device

19.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:lookup-device",
  "id": "",
  "params": {
    "account": "",
    "device-id": ""
  }
}
```

Retrieve information about a given device

This command retrieves the device id, the device type, and the optional device attributes for a given device owned by the provided account.

The format of the device attributes, if present, is specified by the device type the device is assigned to. The device type, when it is created through the `exodm:create-device-type` command, can optionally refer to a yang specification module (created with `exodm:create-yang-module`), that defines the structure of the attributes that all devices of the given type will have.

19.1.1 Elements

account The account, accessible by the authorizing user, to which the device belongs. If no account is given, the authorizing user assumes to have access to only one account, which will be used. If no account is given, and the user has access to multiple accounts, an error is returned. (**type:** string; [mandatory: false])

device-id The ID of the device to look up. The device has previously been created through `exodm:create-device` command. (**type:** string; [mandatory: true])

19.2 Reply

```
{
  "jsonrpc": "2.0",
  "id": "",
  "result": {
    "result": "",
    "device-id": ""
  }
}
```

```

        "device-type": "",
        "attributes": ""
    }
}

```

19.2.1 Elements

attributes Attributes set for this device. The structure of the attributes is defined by the yang module provided to the `exodm:create-device-type` command that created the device type that the device is assigned to. (**type:** XML; [mandatory: false])

device-id Account-wide unique device-id (**type:** string; [mandatory: true])

device-type Type of device, as given by a previous `exodm:create-device-type` call. (**type:** string; [mandatory: true])

result Result of the operation (**type:** enumeration; [mandatory: false])

20 exodm:list-devices

20.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:list-devices",
  "id": "",
  "params": {
    "account": "",
    "n": "",
    "previous": ""
  }
}
```

List all devices created under an account

This command lists the device ID of all devices created under the given account through the **exodm:create-device** command. Detailed information can be given for a device through the **exodm:lookup-device** command.

This command allows for chunks of the result set to be returned to facilitate a sequential retrieval of device IDs.

20.1.1 Elements

account The account, accessible by the authorizing user, to which the devices belong. If no account is given, the authorizing user assumes to have access to only one account, which will be used. If no account is given, and the user has access to multiple accounts, an error is returned. (**type:** string; [mandatory: false])

n Maximum number of entries to return. More devices may be available after the last returned element. (**type:** uint32; [mandatory: true])

previous Start the fetch at the first device after the device with the ID specified by this argument. Use "" if the fetch is to start from the beginning (**type:** string; [mandatory: true])

20.2 Reply

```
{
  "jsonrpc": "2.0",
  "id": "",
  "result": {
    "devices": [
```

```

        {
            "device-id": "",
            "device-type": ""
        }
    ]
}

```

20.2.1 Elements

device-id Account-wide unique device-id (**type:** string; [mandatory: true])

device-type Type of device, as given by a previous [exodm:create-device-type](#) call. (**type:** string; [mandatory: true])

devices Device object data (**type:** array)

21 exodm:delete-devices

21.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:delete-devices",
  "id": "",
  "params": {
    "account": "",
    "device-id": ""
  }
}
```

Delete an existing device

This command deletes a device, previously created with [exodm:create-device](#), from the Exosense Server. This command also deletes the device's attributes, previously set-up through the [exodm:create-device](#) and [exodm:update-device](#) commands.

21.1.1 Elements

account The account, accessible by the authorizing user, to which the device belong. If no account is given, the authorizing user assumes to have access to only one account, which will be used. If no account is given, and the user has access to multiple accounts, an error is returned. (**type:** string; [mandatory: false])

device-id The ID of the device to remove (**type:** string; [mandatory: false])

21.2 Reply

```
{
  "jsonrpc": "2.0",
  "id": "",
  "result": {
    "result": ""
  }
}
```

21.2.1 Elements

result Result of the operation (**type:** enumeration; [mandatory: false])

22 exodm:create-device-group

22.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:create-device-group",
  "id": "",
  "params": {
    "account": "",
    "group-name": "",
    "notification-url": ""
  }
}
```

Create a device group

This command creates a new device group that can have devices added to it. The name of the group must be unique across all device IDs and device groups owned by the given account.

A device group can have devices added to it through the [exodm:add-device-group-members]. Once a group has been created, and devices have been added to it, an RPC command in a package can be invoked on all member devices by sending the RPC to the device group instead of individual devices.

Incoming RPCs from member devices will be sent to the notification URL of all device groups that the device is a member of. See [exodm:add-device-group-members](#) for details.

Packages installed on a group will be pushed out to all devices of that group.

22.1.1 Elements

account The account, accessible by the authorizing user, that will own the device group. If no account is given, the authorizing user assumes to have access to only one account, which will be used. If no account is given, and the user has access to multiple accounts, an error is returned. (**type:** string; [mandatory: false])

group-name Name of the device group. Must be unique across all devices and groups owned by the account. (**type:** string; [mandatory: true])

notification-url Callback to forward device originated RPCs to. (**type:** string; [mandatory: true])

22.2 Reply

```
{
  "jsonrpc": "2.0",
  "id": "",
  "result": {
    "result": ""
  }
}
```

22.2.1 Elements

result Result of the operation (**type:** enumeration; [mandatory: false])

23 exodm:delete-device-group

23.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:delete-device-group",
  "id": "",
  "params": {
    "account": "",
    "group-name": ""
  }
}
```

Delete a device group

This command deletes a device group previously created through a **exodm:create-device-group** command. Any devices added to the group through an **exodm:add-device-group-members** will be removed as members from it prior to the groups deletion.

23.1.1 Elements

account The account, accessible by the authorizing user, that owns the device group to delete. If no account is given, the authorizing user assumes to have access to only one account, which will be used. If no account is given, and the user has access to multiple accounts, an error is returned. (**type:** string; [mandatory: false])

group-name Name of the device group to delete (**type:** string; [mandatory: true])

23.2 Reply

```
{
  "jsonrpc": "2.0",
  "id": "",
  "result": {
    "result": ""
  }
}
```

23.2.1 Elements

result Result of the operation (**type:** enumeration; [mandatory: false])

24 exodm:list-device-groups

24.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:list-device-groups",
  "id": "",
  "params": {
    "account": "",
    "n": "",
    "previous": "",
    "device-id": ""
  }
}
```

List all device groups owned by an account

This command will return all device groups created under an account through the **exodm:create-device-group** command.

This command allows for chunks of the result set to be returned to facilitate a sequential retrieval of device names

24.1.1 Elements

account The account, accessible by the authorizing user, to list the device groups for. If no account is given, the authorizing user assumes to have access to only one account, which will be used. If no account is given, and the user has access to multiple accounts, an error is returned. (**type:** string; [mandatory: false])

device-id Optional device ID. If given, device groups associated with the given device are listed. (**type:** string; [mandatory: false])

n Maximum number of entries to return. More device groups may be available after the last returned element. (**type:** uint32; [mandatory: true])

previous Start the fetch at the first device group after that with the name specified by this argument. Use "" if the fetch is to start from the beginning (**type:** string; [mandatory: true])

24.2 Reply

```
{
  "jsonrpc": "2.0",
```

```

    "id": "",
    "result": {
      "device-groups": [
        {
          "group-name": "",
          "notification-url": ""
        }
      ]
    }
  }
}

```

24.2.1 Elements

device-groups (**type:** array)

group-name The name of the device group (**type:** string; [mandatory: false])

notification-url The notification URL of the device group (**type:** string; [mandatory: false])

25 exodm:list-device-group-members

25.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:list-device-group-members",
  "id": "",
  "params": {
    "account": "",
    "group-name": "",
    "n": "",
    "previous": ""
  }
}
```

List all device members of a device group under an account

This command will list all devices that are a member of the given device group. The listed devices have previously been added to the group through an [exodm:add-device-group-members](#) call.

This command allows for chunks of the result set to be returned to facilitate a sequential retrieval of device names

25.1.1 Elements

account The account, accessible by the authorizing user, that owns the device group. If no account is given, the authorizing user assumes to have access to only one account, which will be used. If no account is given, and the user has access to multiple accounts, an error is returned. (**type:** string; [mandatory: false])

group-name Name of the device group to list members for (**type:** string; [mandatory: true])

n Maximum number of entries to return. More devices may be available after the last returned element. (**type:** uint32; [mandatory: true])

previous Start the fetch at the first devices after that with the device ID specified by this argument. Use "" if the fetch is to start from the beginning (**type:** string; [mandatory: true])

25.2 Reply

```
{
```

```

    "jsonrpc": "2.0",
    "id": "",
    "result": {
        "devices": [
            {
                "device-id": "",
                "primary": ""
            }
        ]
    }
}

```

25.2.1 Elements

device-id The ID of the device (**type:** string; [mandatory: false])

devices (**type:** array)

primary Set to true if this is the primary group for the given device (**type:** “1” (true) | “0” (false); [mandatory: false])

26 exodm:add-device-group-members

26.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:add-device-group-members",
  "id": "",
  "params": {
    "account": "",
    "group-name": "",
    "device-id": [
      "device-id": ""
    ],
    "primary": ""
  }
}
```

Add one or more devices as members of a device group

This command will add a list of devices as members to a single device group. Once this command has been completed, JSON-RPCs sent to the device group will be forwarded to all member devices.

If a device, who is a member of multiple groups, sends an RPC to the Exosense Server, the RPC will be forwarded to the notification URL (as JSON-RPC) of all groups. The reply to send back to the device is read from the device group marked as primary for the device. This allows multiple URLs to receive device data while ensuring that only one reply is sent back to the device.

If a device is only member of a single group, the primary flag is ignored.

If a device itself has a notification URL configured, incoming RPCs from the device will be forwarded to that URL and the notification URL of all device groups that the device is a member of. In these cases, the reply to send back to the device will always be read from the notification URL of the device, thus silently dropping the reply from all device groups that the device is a member of, including the primary group.

If a device is member of a device group, and has it marked as primary, when the **exodm:add-device-group-members** command also has the primary flag set, the primary flag will be moved from the previous group to the new one for the device. Thus, only one group will be marked as primary for a device at any given time.

26.1.1 Elements

account The account, accessible by the authorizing user, that owns the device group and devices. If no account is given, the authorizing user assumes to have access to only one account, which will be used. If no account is given, and the user has access to multiple accounts, an error is returned. (**type:** string; [mandatory: false])

device-id List of device IDs to add as members to the given device group (**type:** string; [mandatory: false])

group-name Name of device group to which to add the given device(s) as members (**type:** string; [mandatory: false])

primary Flag to indicate that this group should be marked as the primary group for all added devices. If any of the added devices already have primary groups setup, those groups will be de-selected in favor of the group specified by this call. (**type:** “1” (true) | “0” (false); [mandatory: false])

26.2 Reply

```
{
  "jsonrpc": "2.0",
  "id": "",
  "result": {
    "result": ""
  }
}
```

26.2.1 Elements

result Result of the operation (**type:** enumeration; [mandatory: false])

27 exodm:remove-device-group-members

27.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:remove-device-group-members",
  "id": "",
  "params": {
    "account": "",
    "group-name": "",
    "device-id": [
      "device-id": ""
    ]
  }
}
```

Remove a list of devices as members from a device group

This command removes the specified devices from the given group. The devices have previously been added to the group through the **exodm:add-device-group-members** command. If a device removed from a group has that group marked as primary, and the device will still be member of more than one group, and the device itself does not have a notification URL setup, then any future RPCs received from that device will result in an error being sent back to it by the Exosense Server. This is due to the fact that Exosense does not know which notification URL to read the reply from.

27.1.1 Elements

account The account, accessible by the authorizing user, that owns the device group and devices. If no account is given, the authorizing user assumes to have access to only one account, which will be used. If no account is given, and the user has access to multiple accounts, an error is returned. (**type:** string; [mandatory: false])

device-id The Device IDs of the devices to remove as members from the device group. (**type:** string; [mandatory: false])

group-name Name of the device group to remove the device(s) from. (**type:** string; [mandatory: false])

27.2 Reply

```
{
```



```
    "jsonrpc": "2.0",
    "id": "",
    "result": {
        "result": ""
    }
}
```

27.2.1 Elements

result Result of the operation (**type:** enumeration; [mandatory: false])

28 exodm:update-device-group

28.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:update-device-group",
  "id": "",
  "params": {
    "account": "",
    "group-name": "",
    "notification-url": ""
  }
}
```

Update a device group

This command will update the notification URL of the given device group. Once this command has completed, any incoming, device-originated RPCs sent to the device group will be forwarded as JSON-RPC to the given URL.

28.1.1 Elements

account The account, accessible by the authorizing user, that owns the device group. If no account is given, the authorizing user assumes to have access to only one account, which will be used. If no account is given, and the user has access to multiple accounts, an error is returned. (**type:** string; [mandatory: false])

group-name Name of the device group to update. (**type:** string; [mandatory: true])

notification-url RPC notification URL to forward for incoming device-originated RPCs to. (**type:** string; [mandatory: true])

28.2 Reply

```
{
  "jsonrpc": "2.0",
  "id": "",
  "result": {
    "result": ""
  }
}
```

28.2.1 Elements

result Result of the operation (**type:** enumeration; [mandatory: false])

29 exodm:create-device-type

29.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:create-device-type",
  "id": "",
  "params": {
    "account": "",
    "name": "",
    "protocol": "",
    "yang-specification": "",
    "attribute-container": ""
  }
}
```

Create a new device type under an account

This command creates a new device type that can be assigned to devices. The type will be owned by an account and use the specified protocol to communicate with devices assigned to the device type. The name of the protocol is one of those registered by protocol plugins deployed on the server.

The device type can optionally be associated with a set of attributes specified by a yang file uploaded through an [exodm:create-yang-module](#) command. In these cases, each device assigned to the device type will have the given attributes associated with it in the server. Unlike device configuration data, device attributes will not be sent to the device. Instead they will be stored locally on the Exosense Server.

The attributes can be set through the [exodm:update-device](#) command and retrieved through [exodm:lookup-device](#). Protocol plugins can also access the device attributes to locate network addresses and other information necessary for the protocol to communicate with a device.

Please note that the actual configuration data and RPC APIs for a device is managed through the packages installed on it.

29.1.1 Elements

account The account, accessible by the authorizing user, to which the device type belongs. If no account is given, the authorizing user assumes to have access to only one account, which will be used. If no account is given, and the user has access to multiple accounts, an error is returned. (**type:** string; [mandatory: false])

attribute-container Container within the Yang specification hosting the device attribute specification. (**type:** string; [mandatory: false])

name Account-wide unique name of the device type. (**type:** string; [mandatory: true])

protocol Protocol between device and Exosense server. The name has previously been registered by a protocol plugin deployed on the server. (**type:** string; [mandatory: true])

yang-specification Yang specification containing the device attribute specifications. (**type:** string; [mandatory: false])

29.2 Reply

```
{
  "jsonrpc": "2.0",
  "id": "",
  "result": {
    "result": ""
  }
}
```

29.2.1 Elements

result Result of the operation (**type:** enumeration; [mandatory: false])

30 exodm:update-device-type

30.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:update-device-type",
  "id": "",
  "params": {
    "account": "",
    "name": "",
    "protocol": "",
    "yang-specification": "",
    "attribute-container": ""
  }
}
```

Update an existing device type

This commands updates an existing device type previously created with [exodm:create-device-type](#). This command currently does nothing since changing any information would have unforeseeable consequences for the existing devices of this type.

30.1.1 Elements

account The account, accessible by the authorizing user, to which the device type belongs. If no account is given, the authorizing user assumes to have access to only one account, which will be used. If no account is given, and the user has access to multiple accounts, an error is returned. (**type:** string; [mandatory: false])

attribute-container Container within the Yang specification hosting the device attribute specification. (**type:** string; [mandatory: false])

name Account-wide unique name of the device type. (**type:** string; [mandatory: true])

protocol Protocol between device and Exosense server. The name has previously been registered by a protocol plugin deployed on the server. (**type:** string; [mandatory: false])

yang-specification Yang specification containing the device attribute specifications. (**type:** string; [mandatory: false])

30.2 Reply

```
{
  "jsonrpc": "2.0",
  "id": "",
  "result": {
    "result": ""
  }
}
```

30.2.1 Elements

result Result of the operation (**type:** enumeration; [mandatory: false])

31 exodm:delete-device-type

31.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:delete-device-type",
  "id": "",
  "params": {
    "account": "",
    "name": ""
  }
}
```

Delete a device type and its associated devices

This command deletes a device type previously created through a **exodm:create-device-type** command. All devices assigned to the device type will also be deleted as if [exodm:delete-device] was called for each one of them.

31.1.1 Elements

account The account, accessible by the authorizing user, to which the device type belongs. If no account is given, the authorizing user assumes to have access to only one account, which will be used. If no account is given, and the user has access to multiple accounts, an error is returned. (**type:** string; [mandatory: false])

name Name of the device type to delete. (**type:** string; [mandatory: true])

31.2 Reply

```
{
  "jsonrpc": "2.0",
  "id": "",
  "result": {
    "result": ""
  }
}
```

31.2.1 Elements

result Result of the operation (**type:** enumeration; [mandatory: false])

32 exodm:list-device-types

32.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:list-device-types",
  "id": "",
  "params": {
    "account": "",
    "n": "",
    "previous": ""
  }
}
```

List all device types owned by an account

This commands will retrieve the name of all device types created under the given account through [exodm:create-device-type](#) calls.

This command allows for chunks of the result set to be returned to facilitate a sequential retrieval of device type names.

32.1.1 Elements

account The account, accessible by the authorizing user, to which the device type belongs. If no account is given, the authorizing user assumes to have access to only one account, which will be used. If no account is given, and the user has access to multiple accounts, an error is returned. (**type:** string; [mandatory: false])

n Maximum number of entries to return. More device types may be available after the last returned element. (**type:** uint32; [mandatory: false])

previous Start the fetch at the first device type after the one with the name specified by this argument. Use "" if the fetch is to start from the beginning (**type:** string; [mandatory: false])

32.2 Reply

```
{
  "jsonrpc": "2.0",
  "id": "",
  "result": {
    "device-types": [
```

```

        {
            "name": "",
            "protocol": "",
            "yang-specification": "",
            "attribute-container": ""
        }
    ]
}

```

32.2.1 Elements

attribute-container Container within the Yang specification hosting the device attribute specification. (**type:** string; [mandatory: false])

device-types List of device types owned by the given account. (**type:** array)

name Account-wide unique name of the device type. (**type:** string; [mandatory: true])

protocol Protocol between device and Exosense server. The name has previously been registered by a protocol plugin deployed on the server. (**type:** string; [mandatory: true])

yang-specification Yang specification containing the device attribute specifications. (**type:** string; [mandatory: false])

33 exodm:list-device-type-devices

33.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:list-device-type-devices",
  "id": "",
  "params": {
    "account": "",
    "name": "",
    "n": "",
    "previous": ""
  }
}
```

List device IDs associated with a device type

This command returns the device ID of all devices that were assigned to the given device type when they were created with a **exodm:create-device** command.

This command allows for chunks of the result set to be returned to facilitate a sequential retrieval of device IDs.

33.1.1 Elements

account The account, accessible by the authorizing user, to which the device type belongs. If no account is given, the authorizing user assumes to have access to only one account, which will be used. If no account is given, and the user has access to multiple accounts, an error is returned. (**type:** string; [mandatory: false])

n Maximum number of entries to return. More devices may be available after the last returned element. (**type:** uint32; [mandatory: true])

name Name of device type for which to retrieve associated devices. (**type:** string; [mandatory: true])

previous Start the fetch at the first device after the device with the ID specified by this argument. Use "" if the fetch is to start from the beginning (**type:** string; [mandatory: true])

33.2 Reply

```
{
  "jsonrpc": "2.0",
```

```
    "id": "",  
    "result": "ok"  
}
```

34 exodm:create-config-set

34.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:create-config-set",
  "id": "",
  "params": {
    "account": "",
    "name": "",
    "yang": "",
    "notification-url": "",
    "values": ""
  }
}
```

Create a new configuration set [deprecated in 2.x]

This command creates a new configuration set describing an RPC API and configuration data supported by a device. A configuration set is associated with a yang module, created through a [exodm:create-yang-module](#) command, that describes the configuration data and API interface of the devices that are members of the set. Also tied to the configuration set is a notification URL that will receive all results from RPCs sent to a member device. The notification URL will also be used to forward incoming incoming RPCs from the device sent to the server.

Once a configuration set has been created, it can have devices added to it through the [exodm:add-config-set-members](#). All such member devices can have RPCs, as defined in the configuration set yang module, called through the JSON-RPC interface. These devices can also have configuration data updated by the server through the [exodm:update-config-set](#) and [exodm:push-config-set](#) commands.

Please note that any RPCs defined in the yang module tied to a configuration set must have a “uses `exo:std-request`” line in it to include transaction elements used by the Exosense Server to route the RPC to the correct device. See [\[exo:std-request\]](#) for details.

34.1.1 Elements

account The account, accessible by the authorizing user, to which the configuration set belongs. If no account is given, the authorizing user assumes to have access to only one account, which will be used. If no account is given, and the user has access to multiple accounts, an error is returned. (**type:** string; [mandatory: false])

name Name of the configuration set to create (**type:** string; [mandatory: true])

notification-url Notification URL to send RPC results and device-originated RPC calls to. (**type:** string; [mandatory: false])

values Configuration values (validated against the yang spec) to initialize the newly created configuration set with. (**type:** XML; [mandatory: false])

yang Name of the yang module to use. Include the “.yang” suffix. (**type:** string; [mandatory: true])

34.2 Reply

```
{
  "jsonrpc": "2.0",
  "id": "",
  "result": {
    "result": ""
  }
}
```

34.2.1 Elements

result Result of the operation (**type:** enumeration; [mandatory: false])

35 exodm:update-config-set

35.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:update-config-set",
  "id": "",
  "params": {
    "name": "",
    "account": "",
    "notification-url": "",
    "values": ""
  }
}
```

Update the data of a configuraiton set

This command will update the configuration values of a configuraiton set. Once the data has been updated, it can be pushed out to all configuration set member devices through the **exodm:push-config-set** command.

The format of the configuration data is specified by the yang module provided to the **exodm:create-config-set** command that created the configuration set. Any data that does not validate against the yang module will result in an error.

35.1.1 Elements

account The account, accessible by the authorizing user, to which the configuration set belongs. If no account is given, the authorizing user assumes to have access to only one account, which will be used. If no account is given, and the user has access to multiple accounts, an error is returned. (**type:** string; [mandatory: false])

name The name of the configuration data set to update. (**type:** string; [mandatory: true])

notification-url Notification URL to send RPC results and device-originated RPC calls to. (**type:** string; [mandatory: false])

values Configuration values (validated against the yang spec) to update in the configuration set. (**type:** XML; [mandatory: false])

35.2 Reply

```
{
```

```
    "jsonrpc": "2.0",
    "id": "",
    "result": {
        "result": ""
    }
}
```

35.2.1 Elements

result Result of the operation (**type:** enumeration; [mandatory: false])

36 exodm:delete-config-set

36.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:delete-config-set",
  "id": "",
  "params": {
    "account": "",
    "name": ""
  }
}
```

Delete an existing configuration set

This command deletes a configuration set previously created through an **exodm:create-config-set** command. Any configuration data pushed to the set's member devices will be delivered, although the delivery notification sent back to the notification URL of the set may be lost. Any configuration set member devices will be removed from the set prior to its deletion.

36.1.1 Elements

account The account, accessible by the authorizing user, to which the configuration set belongs. If no account is given, the authorizing user assumes to have access to only one account, which will be used. If no account is given, and the user has access to multiple accounts, an error is returned. (**type:** string; [mandatory: false])

name The name of the configuration data set to delete. (**type:** string; [mandatory: true])

36.2 Reply

```
{
  "jsonrpc": "2.0",
  "id": "",
  "result": {
    "result": ""
  }
}
```

36.2.1 Elements

result Result of the operation (**type:** enumeration; [mandatory: false])

37 exodm:list-config-sets

37.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:list-config-sets",
  "id": "",
  "params": {
    "account": "",
    "n": "",
    "previous": "",
    "device-id": ""
  }
}
```

List all configuration sets owned by an account

This command will return all configuration sets created under the specified account through a previous **exodm:create-config-set** command.

37.1.1 Elements

account The account, accessible by the authorizing user, to which the configuration set belongs. If no account is given, the authorizing user assumes to have access to only one account, which will be used. If no account is given, and the user has access to multiple accounts, an error is returned. (**type:** string; [mandatory: false])

device-id Optional device ID. If given, config specs associated with the given device are listed. (**type:** string; [mandatory: false])

n Number of entries to fetch (**type:** uint32; [mandatory: true])

previous Previous config set; "" if from beginning (**type:** string; [mandatory: true])

37.2 Reply

```
{
  "jsonrpc": "2.0",
  "id": "",
  "result": {
    "config-sets": [
      {
```

```

        "name": "",
        "yang": "",
        "notification-url": ""
    }
]
}
}

```

37.2.1 Elements

config-sets (**type:** array)

name The name of the configuration set, as specified by the [exodm:create-config-set](#) call that created it. (**type:** string; [mandatory: false])

notification-url The notification URL associated with the configuration set. (**type:** string; [mandatory: false])

yang The name of the yang module associated with the configuration set. (**type:** string; [mandatory: false])

38 exodm:list-config-set-members

38.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:list-config-set-members",
  "id": "",
  "params": {
    "name": "",
    "n": "",
    "previous": "",
    "account": ""
  }
}
```

List all device members of a configuration set

This command will list all devices added as member to the given configuration set through previous **exodm:add-config-set-members** commands.

This command allows for chunks of the result set to be returned to facilitate a sequential retrieval of accounts.

38.1.1 Elements

account The account, accessible by the authorizing user, to which the configuration set belongs. If no account is given, the authorizing user assumes to have access to only one account, which will be used. If no account is given, and the user has access to multiple accounts, an error is returned. (**type:** string; [mandatory: false])

n Maximum number of entries to return. More devices may be available after the last returned element. (**type:** uint32; [mandatory: true])

name Name of the config set to list member devices for (**type:** string; [mandatory: true])

previous Start the fetch at the first device after the device with the id specified by this argument. Use "" if the fetch is to start from the beginning (**type:** string; [mandatory: true])

38.2 Reply

```
{
  "jsonrpc": "2.0",
```

```
    "id": "",
    "result": {
      "config-set-members": [
        "config-set-members": ""
      ]
    }
  }
```

38.2.1 Elements

config-set-members Device ID of devices that are a member of the given configuration set. (**type:** string; [mandatory: false])

39 exodm:add-config-set-members

39.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:add-config-set-members",
  "id": "",
  "params": {
    "account": "",
    "name": [
      "name": ""
    ],
    "device-id": [
      "device-id": ""
    ]
  }
}
```

Add devices as a members to one or more configuration sets

This command adds the listed devices as members of a collection of configuration sets. When this command returns, the RPCs of the yang specification associated with the configuration set through the [exodm:create-config-set](#) command can be invoked for the devices.

The devices will also be able to have the configuration data setup for the configuration set through the [exodm:create-config-set](#) and [exodm:update-config-set](#) pushed to them using the [exodm:push-config-set](#) command.

If a member device replies to an RPC sent to it, or originates an RPC of its own, the JSON-RPC will be sent to the notification URL associated with the configuration set.

39.1.1 Elements

account The account, accessible by the authorizing user, to which the configuration set belongs. If no account is given, the authorizing user assumes to have access to only one account, which will be used. If no account is given, and the user has access to multiple accounts, an error is returned. (**type:** string; [mandatory: false])

device-id Devices to add to the listed configuration sets. (**type:** string; [mandatory: false])

name Name of configuration sets to add devices to as members. (**type:** string; [mandatory: false])

39.2 Reply

```
{
  "jsonrpc": "2.0",
  "id": "",
  "result": {
    "result": ""
  }
}
```

39.2.1 Elements

result Result of the operation (**type:** enumeration; [mandatory: false])

40 exodm:remove-config-set-members

40.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:remove-config-set-members",
  "id": "",
  "params": {
    "account": "",
    "name": [
      "name": ""
    ],
    "device-id": [
      "device-id": ""
    ]
  }
}
```

Remove devices as a members from one or more configuration sets

This command removes a list of devices from one or more configuration sets. The devices have previously been added as configuration set members through an [exodm:add-config-set-members](#) command.

40.1.1 Elements

account The account, accessible by the authorizing user, to which the configuration set belongs. If no account is given, the authorizing user assumes to have access to only one account, which will be used. If no account is given, and the user has access to multiple accounts, an error is returned. (**type:** string; [mandatory: false])

device-id Device IDs to remove as members from the given configuration sets. (**type:** string; [mandatory: false])

name Configuration sets from which to remove the given devices (**type:** string; [mandatory: false])

40.2 Reply

```
{
  "jsonrpc": "2.0",
  "id": "",
  "result": {
```

```
        "result": ""
    }
}
```

40.2.1 Elements

result Result of the operation (**type:** enumeration; [mandatory: false])

41 exodm:push-config-set

41.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:push-config-set",
  "id": "",
  "params": {
    "account": "",
    "name": ""
  }
}
```

Push configuration set data to all member devices

This command will transfer the configuration data in the configuration set to all member devices. The configuration data is setup in the configuration set through the **exodm:create-config-set** and **[exodm:update-config]**. For each member device that receives the configuration set, an **[exodm:push-config-set-callback]** notification will be sent back to the notification URL associated with the configuration set.

41.1.1 Elements

account The account, accessible by the authorizing user, to which the configuration set belongs. If no account is given, the authorizing user assumes to have access to only one account, which will be used. If no account is given, and the user has access to multiple accounts, an error is returned. (**type:** string; [mandatory: false])

name Name of configuration set whose data to push to the member devices.. (**type:** string; [mandatory: false])

41.2 Reply

```
{
  "jsonrpc": "2.0",
  "id": "",
  "result": {
    "result": ""
  }
}
```

41.2.1 Elements

result Result of the operation (**type:** enumeration; [mandatory: false])

41.3 Notification: exodm:push-config-set-callback

```
{
  "jsonrpc": "2.0",
  "method": "exodm:push-config-set-callback",
  "params": {
    "transaction-id": "",
    "rpc-status": "",
    "rpc-status-string": "",
    "final": ""
  }
}
```

The callback made for each device member of a configuration set that receives configuration data through an [exodm:push-config-data] command.

41.3.1 Elements

final This is the final callback for the given operation (**type:** “1” (true) | “0” (false); [mandatory: false])

rpc-status Status of operation in progress. (**type:** enumeration; [mandatory: false])

rpc-status-string Additional status information, in human readable format, (**type:** string; [mandatory: false])

transaction-id The transaction that this callback is made in response to. (**type:** uint32; [mandatory: false])

42 exodm:create-yang-module

42.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:create-yang-module",
  "id": "",
  "params": {
    "account": "",
    "repository": "",
    "name": "",
    "yang-module": ""
  }
}
```

Upload a yang file

This commands parses the provided yang file and stores it as a yang module in the Exosense Server. The yang file is expected to be compliant with RFC 6020. Once a yang module has been created, it can be used to specify the structure of device attributes, package RPC APIs, and package configuration data. Please see [exodm:create-device-type](#) for details on device attributes, and [exodm:create-package](#) for details on package RPC API and configuration data structure.

A yang module can be created either in the account or system repository on the Exosense Server. Modules created in the “system repository will be accessible to all accounts on the system, while modules created in the”account” repository will only be visible and accessible to the account invoking this command. Only the sysadm account can create a yang module in the system repository.

If the yang module is uploaded to the “user” repo, then the name of the yang module must be unique among all existing yang modules under the given account, as well as being unique among all existing yang modules in the “system” repository.

If the yang module is uploaded to the “system” repo, then the name of the yang module must be unique among all existing yang modules created in the Exosense Server. No existing yang module in any “user” repository under any account can have the same name as the new module. Neither can the new module have the same name as an existing yang module in the “system” repository.

42.1.1 Elements

account The account, accessible by the authorizing user, in whose repository the yang modules should be created.If no account is given, the authorizing user

assumes to have access to only one account, which will be used. If no account is given, and the user has access to multiple accounts, an error is returned.

This argument is not used if the yang module to is to be created in the “system” repository. (**type:** string; [mandatory: false])

name Name of the uploaded yang module, including the .yang prefix. Example is commands.yang. (**type:** string; [mandatory: true])

repository Where to store the module in the Exosense Server: Can be “system” or “account” (**type:** string; [mandatory: false])

yang-module The yang code to create the module from (**type:** string; [mandatory: true])

42.2 Reply

```
{
  "jsonrpc": "2.0",
  "id": "",
  "result": {
    "result": ""
  }
}
```

42.2.1 Elements

result Result of the operation (**type:** enumeration; [mandatory: false])

43 exodm:delete-yang-module

43.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:delete-yang-module",
  "id": "",
  "params": {
    "account": "",
    "name": ""
  }
}
```

Delete an existing yang module

This command deletes a yang module previously created through **exodm:create-yang-module**. This command will fail if there are any device types and/or packages currently referring to the given yang module.

43.1.1 Elements

account The account, accessible by the authorizing user, to which the yang modules belongs. If no account is given, the authorizing user assumes to have access to only one account, which will be used. If no account is given, and the user has access to multiple accounts, an error is returned.

This argument is not used if the yang module to delete resides in the “system” repository. (**type:** string; [mandatory: false])

name Name of the yang module to delete. Example iscommands.yang’ (**type:** string; [mandatory: true])

43.2 Reply

```
{
  "jsonrpc": "2.0",
  "id": "",
  "result": {
    "result": ""
  }
}
```

43.2.1 Elements

result Result of the operation (**type:** enumeration; [mandatory: false])

44 exodm:list-yang-modules

44.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:list-yang-modules",
  "id": "",
  "params": {
    "account": "",
    "repository": "",
    "n": "",
    "previous": ""
  }
}
```

List the yang modules created in a repository This command lists all yang modules under the account-specific or system-wide repository.

This command allows for chunks of the result set to be returned to facilitate a sequential retrieval of device IDs.

44.1.1 Elements

account The account, accessible by the authorizing user, to which the yang modules belongs. If no account is given, the authorizing user assumes to have access to only one account, which will be used. If no account is given, and the user has access to multiple accounts, an error is returned.

This argument is not used if the yang modules to list resides in the “system” repository. (**type:** string; [mandatory: false])

n Maximum number of entries to return. More modules may be available after the last returned element. (**type:** uint32; [mandatory: true])

previous Start the fetch at the first module after the module with the name specified by this argument. Use "" if the fetch is to start from the beginning (**type:** string; [mandatory: true])

repository The repository to list modules under “system” or “account”; default: “account” (**type:** string; [mandatory: false])

44.2 Reply

```
{
  "jsonrpc": "2.0",
```

```

    "id": "",
    "result": {
      "yang-modules": [
        "yang-modules": ""
      ]
    }
  }
}

```

44.2.1 Elements

yang-modules List of yang modules. (**type:** string; [mandatory: false])

45 exodm:create-package

45.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:create-package",
  "id": "",
  "params": {
    "account": "",
    "package-name": "",
    "prerequisite-packages": [
      "prerequisite-packages": ""
    ],
    "upgrade-from-packages": [
      "upgrade-from-packages": ""
    ],
    "yang-specification": "",
    "configuration-container": "",
    "image": ""
  }
}
```

Upload a package, with a software image

This command uploads a package software image and the associated meta data to the Exosense Server. Once the package has been uploaded, it can be added to device types that supports it through the [exodm:add-package-to-device-type](#) command, and then installed on actual devices through the [exodm:install-package](#) command.

If the given package is an upgrade, all existing packages that this package can perform an upgrade on can be listed by the upgrade-from-packages argument. Upgrade packages can be installed on a device through the [exodm:upgrade-package](#)

If the given package needs other packages installed on a device before itself can be installed, the list of the required packages can be listed in the prerequisite-packages argument.

When [exodm:install-package](#) command is called, the server will check its device database to ensure that the prerequisite packages are already installed. If this is not the case, and recursive-dependency-resolve installation has not been selected, an error will be returned. If recursive installation has been selected for the command, all dependency packages will be resolved recursively and transmitted with the package given in package-name to the device.

If a package has functions that can be remotely invoked on a device from the Exosense Server, the API of those RPCs is specified by the yang module given in the yang-specification argument. Please note that all yang RPC definitions are global and cannot be hosted in a container. Once the package has been installed on a device, a user with access to an RPC given in the yang module can call the function on the device through the JSON-RPC interface on the Exosense Server. The server will store the call and forward it to the device once it comes online. The result of the executed function will be transported back from the device to the backend server through a notification sent to the device notification URL and the notification URL of any device groups that the device is a member of.

Please note that any RPCs defined in the yang module tied to a package must have a “uses exo:std-request” line in it to include transaction elements used by the Exosense Server to route the RPC to the correct device. See [exo:std-request] for details.

In order for a package RPC to be executed via a JSON-RPC call, a user must be assigned to a role with execution rights to the RPC. A role is given execution rights to a specific RPC in a package created in an account through the `exodm:add-role-execution-permission` call. Users can then be added to the role through the `exodm:add-user-to-role` the

If a package has configuration data that can be remotely updated from the Exosense Server, the structure of that data is also specified in the yang module given by the yang-specification argument. The container hosting the configuration data structure is given in the configuration-container argument.

The configuration data for a package can be set for a package, once it has been installed on a device, through the [exodm:update-package-configuration] command.

45.1.1 Elements

account The account, accessible by the authorizing user, that will own the package. If no account is given, the authorizing user assumes to have access to only one account, which will be used. If no account is given, and the user has access to multiple accounts, an error is returned. (**type:** string; [mandatory: false])

configuration-container Container within the Yang specification that has the configuration data definition. (**type:** string; [mandatory: false])

image The package image itself. (**type:** XML; [mandatory: false])

package-name Name of package. (**type:** string; [mandatory: false])

prerequisite-packages List of packages that must be installed prior to this on a device (**type:** string; [mandatory: false])

upgrade-from-packages List of packages that can be upgraded by this package.
(**type:** string; [mandatory: false])

yang-specification Yang specification containing the API and configuration data definitions. (**type:** string; [mandatory: false])

45.2 Reply

```
{
  "jsonrpc": "2.0",
  "id": "",
  "result": {
    "result": ""
  }
}
```

45.2.1 Elements

result Result of the operation (**type:** enumeration; [mandatory: false])

46 exodm:add-package-to-device-type

46.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:add-package-to-device-type",
  "id": "",
  "params": {
    "account": "",
    "package-name": [
      "package-name": ""
    ],
    "device-type": "",
    "install-arguments": "",
    "upgrade-arguments": "",
    "uninstall-arguments": ""
  }
}
```

Specify that a package can be installed on devices of a given device type

This command specifies that the given package can be installed, upgraded, and removed from devices that belong to the provided device type. Once the package has been added to the device type, it can be sent out to be installed, upgraded, or uninstalled to/from a device using the `exodm:install-package`, `exodm:upgrade-package`, and `exodm:uninstall-package` calls. In addition to the device type and package name itself, this call also accepts arguments to be provided as command line parameters to the local package manager on the device. Different arguments are given for the manager depending on if the package is to be installed, upgraded or uninstalled. The actual syntax and semantics of the arguments are dependent on the type of package manager running on the device.

46.1.1 Elements

account The account, accessible by the authorizing user, to which the package and device type belong. If no account is given, the authorizing user assumes to have access to only one account, which will be used. If no account is given, and the user has access to multiple accounts, an error is returned. (**type:** string; [mandatory: false])

device-type Device type to accept this package. The device type is that assigned to the device type through a `exodm:create-device-type` command. (**type:** string; [mandatory: false])

install-arguments Arguments to be provided to the device package manager when package is installed on devices of the given device type. The specified string will be supplied as a command line argument to the package manager. The exact syntax of the string is dependent on the package manager running on the device. (**type:** string; [mandatory: false])

package-name The name of the package to allow on the device type. The name is that assigned to the package when it was created through a **exodm:create-package** command. (**type:** string; [mandatory: false])

uninstall-arguments Arguments to be provided to the device package manager when package is uninstalled from devices of the given device type. The specified string will be supplied as a command line argument to the package manager. The exact syntax of the string is dependent on the package manager running on the device. (**type:** string; [mandatory: false])

upgrade-arguments Arguments to be provided to the device package manager when package is upgraded on devices of the given device type. The specified string will be supplied as a command line argument to the package manager. The exact syntax of the string is dependent on the package manager running on the device. (**type:** string; [mandatory: false])

46.2 Reply

```
{
  "jsonrpc": "2.0",
  "id": "",
  "result": {
    "result": ""
  }
}
```

46.2.1 Elements

result Result of the operation (**type:** enumeration; [mandatory: false])

47 exodm:remove-package-from-device-type

47.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:remove-package-from-device-type",
  "id": "",
  "params": {
    "account": "",
    "package-name": [
      "package-name": ""
    ],
    "device-type": "",
    "install-arguments": "",
    "upgrade-arguments": "",
    "uninstall-arguments": ""
  }
}
```

Specify that a package cannot any longer be installed on devices of a given device type

This commands specifies that the given package cannot be installed, upgraded, or removed from devices that belongs to the provided device type. The given package must have been added to the device-type with a previous **exodm:add-package-to-device-type** command.

Any devices of the given device type that already have the package installed will not be affected, although they cannot be uninstalled from them once the package is no longer supported by the device type.

47.1.1 Elements

account The account, accessible by the authorizing user, to which the package and device type belong. If no account is given, the authorizing user assumes to have access to only one account, which will be used. If no account is given, and the user has access to multiple accounts, an error is returned. (**type:** string; [mandatory: false])

device-type Device type to accept this package. The device type is that assigned to the device type through a **exodm:create-device-type** command. (**type:** string; [mandatory: false])

install-arguments Arguments to be provided to the device package manager when package is installed on devices of the given device type. The specified

string will be supplied as a command line argument to the package manager. The exact syntax of the string is dependent on the package manager running on the device. (**type:** string; [mandatory: false])

package-name The name of the package to allow on the device type. The name is that assigned to the package when it was created through a **exodm:create-package** command. (**type:** string; [mandatory: false])

uninstall-arguments Arguments to be provided to the device package manager when package is uninstalled from devices of the given device type. The specified string will be supplied as a command line argument to the package manager. The exact syntax of the string is dependent on the package manager running on the device. (**type:** string; [mandatory: false])

upgrade-arguments Arguments to be provided to the device package manager when package is upgraded on devices of the given device type. The specified string will be supplied as a command line argument to the package manager. The exact syntax of the string is dependent on the package manager running on the device. (**type:** string; [mandatory: false])

47.2 Reply

```
{
  "jsonrpc": "2.0",
  "id": "",
  "result": {
    "result": ""
  }
}
```

47.2.1 Elements

result Result of the operation (**type:** enumeration; [mandatory: false])

48 exodm:install-package

48.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:install-package",
  "id": "",
  "params": {
    "account": "",
    "package-name": "",
    "targets": [
      "targets": ""
    ],
    "recursive-dependency-resolve": "",
    "rpc-destination-url": "",
    "timeout": ""
  }
}
```

Install a package on one or more devices

Installs the given package on the devices provided in the targets argument. The device IDs can either be listed directly in the argument, or be a member of a listed device group. The package will be transmitted to all resolved target and then installed on them.

Please note that all devices, either listed directly or through a device group, must be of a device type that accepts the given package. If one or more devices are of a type not accepting the package, a [exodm:package-operation-notification] will be sent back for the failed device with the error code not-supported.

The installation is carried out by invoking the local package manager on the device. This manager will be provided with command line arguments retrieved from the install-arguments string specified to the [exodm:add-package-to-device-type](#) when the package was added to the device type.

The package to install may have dependencies on other packages that must be installed on a device before itself can be installed. Such dependencies are listed in the prerequisite-packages argument of the [exodm:create-package](#) command that created the package.

Prior to processing the install operation on each device, server will check its internal database to ensure that the prerequisite packages are already installed on the device. If this is not the case, and recursive installation has not been selected, an error will be returned. If the recursive-dependency-resolve argument has been set for the install, all dependency packages will be resolved recursively

and transmitted with the package given in package-name to the device in order to all be installed at once. on the device.

For each successful or failed transmit and install operation on a device, a notification will be sent back to the backend server. This allows the backend server to keep track of successes and failures during the install process. The notification will be sent to the URL registered for the device and any device groups that the device is a member of. Thus, a single device operation may trigger notifications sends to multiple notification URLs. If the same URL is registered multiple times for a device and its groups, the notificaiton will only be sent once to that URL.

48.1.1 Elements

account The account, accessible by the authorizing user, to which the package and devices belong. If no account is given, the authorizing user assumes to have access only one account, which will be used. If no account is given, and the user has access to multiple accounts, an error is returned. (**type:** string; [mandatory: false])

package-name The name of the package to install (**type:** string; [mandatory: true])

recursive-dependency-resolve Recursively transmit and install any packages that this package is dependent upon. This allows a complete tree of software to be transmitted with a single transaciton. (**type:** "1" (true) | "0" (false); [mandatory: false])

rpc-destination-url The destination URL used for incoming RPCs that originate from this package in device. (**type:** string; [mandatory: false])

targets Mix of device-ids and device group ids for all devices to have the package installed. (**type:** string; [mandatory: false])

timeout How many seconds to wait for the package to be installed on the target devices before returning a timeout error. Default is one week. (**type:** uint32; [mandatory: false])

48.2 Reply

```
{
  "jsonrpc": "2.0",
  "id": "",
  "result": {
    "result": ""
  }
}
```

48.2.1 Elements

result Result of the operation (**type:** enumeration; [mandatory: false])

49 exodm:upgrade-package

49.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:upgrade-package",
  "id": "",
  "params": {
    "account": "",
    "from-package-name": "",
    "to-package-name": "",
    "targets": [
      "targets": ""
    ],
    "recursive-dependency-resolve": "",
    "remove-unused": "",
    "timeout": ""
  }
}
```

Upgrade a package on one or more devices

Upgrades the given package, specified by from-package-name, to a newer version, specified in to-package-name, on the devices provided in the targets argument. The device IDs can either be listed directly in the argument, or be a member of a listed device group. The package in to-package-name will be transmitted to all resolved target and then installed on them.

In order for the upgrade to processed, the package specified by the from-package-name must be listed in the upgrade-from-packages element of the **exodm:create-package** call that created the package specified by to-package-name.

Please note that all devices, either listed directly or through a device group, must be of a device type that accepts the given package. If one or more devices are of a type not accepting the package, a [exodm:package-operation-notification] will be sent back for the failed device with the error code not-supported.

The upgrade is carried out by invoking the local package manager on the device. This software will be provided with command line argumens retrieved from the upgrade-arguments string specified to the **exodm:add-package-to-device-type** when the package was added to the device type.

For each successful or failed transmit and install operation on a device, a notification will be sent back to the backend server. This allows the backend server to keep track of successes and failures during the upgrade process. The notification will be sent to the URL registered for the device and any device groups that the device is a member of. Thus, a single device operation may

trigger notifications sends to multiple notification URLs. If the same URL is registered multiple times for a device and its groups, the notification will only be sent once to that URL.

49.1.1 Elements

account The account, accessible by the authorizing user, to which the package and devices belong. If no account is given, the authorizing user assumes to have access to only one account, which will be used. If no account is given, and the user has access multiple accounts, an error is returned. (**type:** string; [mandatory: false])

from-package-name The name of the package already installed on the devices to upgrade. (**type:** string; [mandatory: true])

recursive-dependency-resolve Recursively transmit and install any packages that to-package-name is dependent upon. (**type:** "1" (true) | "0" (false); [mandatory: false])

remove-unused Remove any package images not needed after the upgrade from the device. (**type:** "1" (true) | "0" (false); [mandatory: false])

targets Mix of device-ids and device group ids for all devices to have the upgrade performed. (**type:** string; [mandatory: false])

timeout How many seconds to wait for the package to be upgraded on the target devices before returning a timeout error. Default is one week. (**type:** uint32; [mandatory: false])

to-package-name The name of the package to use to upgrade the from-package. (**type:** string; [mandatory: true])

49.2 Reply

```
{
  "jsonrpc": "2.0",
  "id": "",
  "result": {
    "result": ""
  }
}
```

49.2.1 Elements

result Result of the operation (**type:** enumeration; [mandatory: false])

50 exodm:uninstall-package

50.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:uninstall-package",
  "id": "",
  "params": {
    "account": "",
    "package-name": "",
    "targets": [
      "targets": ""
    ],
    "remove-unused": "",
    "timeout": ""
  }
}
```

Uninstall a package on one or more devices

Uninstalls the given package, specified by `package-name`, from the devices provided in the `targets` argument. The device IDs can either be listed directly in the argument, or be a member of a listed device group.

The uninstall is carried out by invoking the local package manager on the device. This manager will be provided with command line arguments retrieved from the `upgrade-arguments` string specified to the [exodm:add-package-to-device-type](#) when the package was added to the device type.

For each successful or failed uninstall operation on a device, a notification will be sent back to the backend server. This allows the backend server to keep track of successes and failures during the uninstall process. The notification will be sent to the URL registered for the device and any device groups that the device is a member of. Thus, a single device operation may trigger notifications sends to multiple notification URLs. If the same URL is registered multiple times for a device and its groups, the notificaiton will only be sent once to that URL.

50.1.1 Elements

account The account, accessible by the authorizing user, to which the package and devices belong. If no account is given, the authorizing user assumes to have access to only one account, which will be used. If no account is given, and the user has access to multiple accounts, an error is returned. (**type:** string; [mandatory: false])

package-name The name of the package, already installed on the device, to uninstall. (**type:** string; [mandatory: true])

remove-unused Remove the package images that are no longer in use from the device. (**type:** "1" (true) | "0" (false); [mandatory: false])

targets Mix of device-ids and device group ids for all devices to have the uninstall performed. (**type:** string; [mandatory: false])

timeout How many seconds to wait for the package to be removed from the target devices before returning a timeout error. Default is one week. (**type:** uint32; [mandatory: false])

50.2 Reply

```
{
  "jsonrpc": "2.0",
  "id": "",
  "result": {
    "result": ""
  }
}
```

50.2.1 Elements

result Result of the operation (**type:** enumeration; [mandatory: false])

50.3 Notification: exodm:package-operation-notification

```
{
  "jsonrpc": "2.0",
  "method": "exodm:package-operation-notification",
  "params": {
    "device-id": "",
    "package-name": "",
    "package-status": "",
    "transaction-id": "",
    "rpc-status": "",
    "rpc-status-string": "",
    "final": ""
  }
}
```

A notification sent back from the Exosense Server to the backend server to report the progress of an install/upgrade/uninstall operation. One notification will

be sent back for each device that is to process the operation. The device ID specifies the device that the notification refers to. This device ID is listed, either directly or as a device group member, in the targets list of the install-package command that this is a notification of.

50.3.1 Elements

device-id The device that this notification is in response to (**type:** string; [mandatory: false])

final This is the final callback for the given operation (**type:** “1” (true) | “0” (false); [mandatory: false])

package-name The device that this notification is in response to (**type:** string; [mandatory: false])

package-status The status of the package install operation (**type:** “0” (transmitted) | “0” (transmitted) | “1” (transmit-failed) | “2” (installed) | “3” (install-failed) | “4” (upgraded) | “5” (upgrade-failed) | “6” (uninstalled) | “7” (uninstall-failed) | “8” (not-installed) | “9” (unsupported-package) | “10” (dependency-failure) | “11” (timeout); [mandatory: false])

rpc-status Status of operation in progress. (**type:** enumeration; [mandatory: false])

rpc-status-string Additional status information, in human readable format, (**type:** string; [mandatory: false])

transaction-id The transaction that this callback is made in response to. (**type:** uint32; [mandatory: false])

51 exodm:list-packages

51.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:list-packages",
  "id": "",
  "params": {
    "account": "",
    "n": "",
    "previous": ""
  }
}
```

List all packages owned by a given account

This command returns the names of all packages created by previous **exodm:create-package** calls in the given account.

This command allows for chunks of the result set to be returned to facilitate a sequential retrieval of package names.

51.1.1 Elements

account The account, accessible the authorizing user, to list all packages for. If no account is given, the authorizing user assumes to have access to only one account, which will be used. If no account is given, and the user has access to multiple accounts, an error is returned. (**type:** string; [mandatory: false])

n Maximum number of entries to return. More packages may be available after the last returned element. (**type:** uint32; [mandatory: false])

previous Start the fetch at the first package after the package with the name specified by this argument. Use "" if the fetch is to start from the beginning (**type:** uint32; [mandatory: false])

51.2 Reply

```
{
  "jsonrpc": "2.0",
  "id": "",
  "result": {
    "result": "",
    "packages": [
```

```

        "packages": ""
    ]
}

```

51.2.1 Elements

packages Package names under the given account (**type:** string; [mandatory: false])

result Result of the operation (**type:** enumeration; [mandatory: false])

52 exodm:list-device-types-with-package

52.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:list-device-types-with-package",
  "id": "",
  "params": {
    "account": "",
    "package-name": "",
    "n": "",
    "previous": ""
  }
}
```

List all device types that can install a given package

All devices types that have had the given package added to it through an [exodm:add-package-to-device-type](#) command will returned by this call.

This command allows for chunks of the result set to be returned to facilitate a sequential retrieval of device type names.

52.1.1 Elements

account The account, accessible by the authorizing user, to which the package belongs. If no account is given, the authorizing user assumes to have access to only one account, which will be used. If no account is given, and the user has access to multiple accounts, an error is returned. (**type:** string; [mandatory: false])

n Maximum number of entries to return. More device types may be available after the last returned element. (**type:** uint32; [mandatory: false])

package-name The package to which list supporting device types for. (**type:** string; [mandatory: false])

previous Start the fetch at the first device type after that with the name specified by this argument. Use "" if the fetch is to start from the beginning (**type:** uint32; [mandatory: false])

52.2 Reply

```
{
  "jsonrpc": "2.0",
```

```

    "id": "",
    "result": {
        "result": "",
        "device-types": [
            "device-types": ""
        ]
    }
}

```

52.2.1 Elements

device-types List of device types accepting the given package. (**type:** string; [mandatory: false])

result Result of the operation (**type:** enumeration; [mandatory: false])

53 exodm:list-packages-on-device-type

53.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:list-packages-on-device-type",
  "id": "",
  "params": {
    "account": "",
    "device-type": "",
    "n": "",
    "previous": ""
  }
}
```

List all packages that can be installed on the given device type

All packages that have been added to a device type through an **exodm:add-package-to-device-type** command will returned by this call.

This command allows for chunks of the result set to be returned to facilitate a sequential retrieval of device type names.

53.1.1 Elements

account The account, accessible by the authorizing user, to which the device type belongs. If no account is given, the authorizing user assumes to have access to only one account, which will be used. If no account is given, and the user has access to multiple accounts, an error is returned. (**type:** string; [mandatory: false])

device-type The device type to which list supporting device types for. (**type:** string; [mandatory: false])

n Maximum number of entries to return. More package names may be available after the last returned element. (**type:** uint32; [mandatory: false])

previous Start the fetch at the first package after that with the name specified by this argument. Use "" if the fetch is to start from the beginning (**type:** uint32; [mandatory: false])

53.2 Reply

```
{
  "jsonrpc": "2.0",
```

```

    "id": "",
    "result": {
        "result": "",
        "packages": [
            "packages": ""
        ]
    }
}

```

53.2.1 Elements

packages List of package names available for installation on the given device type. (**type:** string; [mandatory: false])

result Result of the operation (**type:** enumeration; [mandatory: false])

54 exodm:list-devices-with-package

54.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:list-devices-with-package",
  "id": "",
  "params": {
    "account": "",
    "package-name": "",
    "n": "",
    "previous": ""
  }
}
```

List all devices that have installed or uninstalled a package

Packages returned have previously been installed on the device using a [exodm:add-package-to-device] call.

Each returned device will have a package-status argument indicating if the package is in the process of being transmitted and installed on a device, or if its installation has completed.

This command allows for chunks of the result set to be returned to facilitate a sequential retrieval of device names

54.1.1 Elements

account The account, accessible by the authorizing user, to which the package belongs. If no account is given, the authorizing user assumes to have access to only one account, which will be used. If no account is given, and the user has access to multiple accounts, an error is returned. (**type:** string; [mandatory: false])

n Maximum number of entries to return. More devices may be available after the last returned element. (**type:** uint32; [mandatory: false])

package-name The account to which list installed devices for. (**type:** string; [mandatory: false])

previous Start the fetch at the first device after that with the ID specified by this argument. Use "" if the fetch is to start from the beginning (**type:** uint32; [mandatory: false])

54.2 Reply

```
{
  "jsonrpc": "2.0",
  "id": "",
  "result": {
    "result": "",
    "devices": [
      {
        "device-id": "",
        "package-status": ""
      }
    ]
  }
}
```

54.2.1 Elements

device-id (**type:** string; [mandatory: false])

devices (**type:** array)

package-status (**type:** “0” (pending-install) | “0” (pending-install) | “1” (installed); [mandatory: false])

result Result of the operation (**type:** enumeration; [mandatory: false])

55 exodm:update-package-config

55.1 Request

```
{
  "jsonrpc": "2.0",
  "method": "exodm:update-package-config",
  "id": "",
  "params": {
    "account": "",
    "targets": [
      "targets": ""
    ],
    "package-name": "",
    "timeout": "",
    "values": ""
  }
}
```

Set and push configuration data for a package to a device(s)

The device IDs can either be listed directly in the argument, or be a member of a listed device group. The configuration data for the package will be transmitted to all resolved target and then updated in their local configuration system. The package name is given together with an array of key/value pairs for the configuration data to set for the given package on the given devices. Each key/value pair must match an element in the yang file and container specified by the create-package yang-specification and configuration-container. All given devices must have both be supporting the specified package, and have the package installed on them.

A successful configuration entry update on a device will overwrite any earlier values that entry had. This allows default and/or general configuration updates, such as Exosense Server addresses and call in schedules, to be sent out to device groups covering large swaths of devices, followed by more targeted updates to individual devices with information such as local encryption keys, individual identities, etc.

For each successful or failed configuration operation on a device, a notification will be sent back to the invoking backend server for the given device. This allows the backend server to keep track of successes and failures during the configuration update process.

55.1.1 Elements

account The account, accessible by the authorizing user, to which the package and device belongs. If no account is given, the authorizing user assumes to have

access to only one account, which will be used. If no account is given, and the user has access multiple accounts, an error is returned. (**type:** string; [mandatory: false])

package-name The package to set the config data for in the target device. (**type:** string; [mandatory: false])

targets A list of device ids and device group ids id to set the config data on. (**type:** string; [mandatory: false])

timeout How many seconds to wait for the configuration data to be sent to the device before returning a timeout error. Default is one week. (**type:** uint32; [mandatory: false])

values Configuration values (validated against the yang specification) to set on the target devices. (**type:** XML; [mandatory: false])

55.2 Reply

```
{
  "jsonrpc": "2.0",
  "id": "",
  "result": "ok"
}
```