

Module: account Service: accountModule DescriptionService DescriptionGET accountsGET account/xPUT account/xGET account/x/accountsPOST account/x/installationDELETE account/x/installation/yGET account/x/devicesPUT account/x/device/yGET account/x/usersGET account/x/installationsDELETE account/x/user/yGET account/x/installation/yPUT account/x/installation/yGET account/x/accounts/countPUT account/x/device/y/nameModule: account Service: installationModule DescriptionService DescriptionPUT installation/x/attributeModule: account Service: UserModule DescriptionService DescriptionPUT user/xGET user/xModule: autha Service: authModule DescriptionService DescriptionGET username/xPOST usernamePOST accountGET account/xGET app/x/tokenGET user/x/token/refreshPOST user/x/logout

Module: authd Service: authModule DescriptionService DescriptionGET device/x/zwavelocaleGET device/x/serversGET device/x/permissionsGET device/x/checklogout/yPOST device/x/logoutGET devicesModule: device Service: deviceModule DescriptionService DescriptionGET device/x/zwavelocaleGET devices_countGET device/x/kitGET device/x/statusModule: eula Service: eulaModule DescriptionService DescriptionGET eulaPOST eulaPUT eulaPOST eula/acceptModule: event Service: eventModule DescriptionService DescriptionGET eventsGET event/xPUT event/xGET data_usageModule: firmware Service: buildModule DescriptionService DescriptionPUT firmwareupgradeModule: firmware Service: firmwareModule DescriptionService Description

[GET firmware](#)[POST firmware](#)[PUT firmware](#)[Module: info Service: config](#)[Module Description](#)[Service Description](#)[POST refresh](#)[GET config](#)[Module: info Service: session](#)[Module Description](#)[Service Description](#)[GET token](#)[GET session/x](#)[Module: locator Service: locator](#)[Module Description](#)[Service Description](#)[GET locator](#)[GET device/x/servers](#)[GET macaddress/x/servers](#)[GET oem/x/servers](#)[Module: relay Service: relay](#)[Module Description](#)[Service Description](#)[GET device/x/usage](#)[POST relay/device/x](#)[POST uiredirect/device/x](#)[PUT account/x/limitusage](#)[GET account/x/limitusage](#)[Module: storage Service: storage](#)[Module Description](#)[Service Description](#)[POST store](#)[GET store/x](#)[GET store/x/archive](#)[GET store/x/thumbnail](#)[GET store](#)[DELETE store/x](#)

[Module: techsupport](#) [Service: techsupport](#)

[Module Description](#)

[Service Description](#)

[POST device/x](#)

Module: [account](#) Service: [account](#)

Module Description

Handles account information and personal information for accounts, users, such as names and addresses.

Service Description

Requests related to managing accounts and associating accounts with users, installations, devices

Module: [account](#) Service: [account](#)

GET accounts

/accounts/accounts/accounts

Description:

Searches for an account based on some criteria

Arguments:

All arguments are optional and used to filter the results

PK_Account_Parent: Only accounts that are children of this account

PK_AccountType: Only accounts of this type

FirstName, LastName, Address1, Address2, City, State, Zip, PK_Country: Searches in the billing address

Phone: Searches in the phone numbers of both accounts and users

Sort: Asc or Desc

Count: count of items (default 50)

Start: items starting count

Response: **Success Code** 200

```
{
  "Accounts": [{
    "PK_Account": 1,
    "PK_AccountType": 5,
    "PK_Oem_Account": "12345",
    "PK_Account_Parent": 2,
    "PK_Account_Parent2": 3,
```

```

"PK_Account_Parent3": 4,
"PK_Account_Parent4"
}}
}

```

Notes:

Only children of the authenticated account will be returned.

Implementation Instructions:
Known issues
Foreign Requests:
Tables:

account	Account
account	Address
account	Phone
account	Account_Phone

Permissions:

lookup_account	Can search for accounts as long as they're children of the authenticated account
----------------	--

Module: [account](#) Service: [account](#)

GET account/x

/accounts/accounts/accounts/x

Description:

Get the details for PK_Account x.

Arguments:

Response:	Success Code	200
------------------	---------------------	------------

```

{
  "PK_AccountType": 1,
  "PK_Oem": 1,
  "Server_Event": "cms1.mios.com",
  "Server_Event_Alt": "cms2.mios.com",
  "Server_Account": "account1.mios.com",

```

```
"Server_Account_Alt": "account2.mios.com",
"PK_Language": 1,
"PK_Oem_Account": "12345",
"Billing": {
  "FirstName": "John",
  "LastName": "Doe",
  "Company": "Acme",
  "Address1": "123 Main St",
  "Address2": "",
  "City": "Suite 100",
  "State": "New York",
  "PostalCode": "NY",
  "PK_Country": 10
},
"Mailing": {
  "FirstName": "Mary"
},
"Phone": [{
  "PK_Provider": 1,
  "PK_Country": 1,
  "Phone": "310-500-1000",
  "Extension": "44",
  "Type": "x",
  "ValidationCode": 12345,
  "Validated": "1"
}, {
  "Phone": "310-500-5000",
  "Extension": "1234",
  "ValidationCode": "1272818",
  "Validated": "0"
}]
```

Tables:

account	Account
account	Address
account	Phone
account	Account_Phone

Permissions:

view_account	Can view basic account information (address, phone, etc.), but not billing
--------------	--

Module: [account](#) Service: [account](#)

PUT [account/x](#)

/account/account/account/x

Description:

x is an account. PUT the account data you want to change in JSON format:

Arguments:

The same JSON data as with the GET:

PK_Account_Parent, PK_AccountType, PK_Oem, PK_Server_Event, PK_Server_Account

Update only whatever information is provided

Response:	Success Code	200
-----------	--------------	-----

```
{
  "PK_AccountType": 1,
  "PK_Oem": 1,
  "PK_Server_Event": 8,
  "PK_Server_Account": 6,
  "PK_Language": 1,
  "PK_Oem_Account": "12345",
  "Billing": {
    "FirstName": "John",
    "LastName": "Doe",
    "Company": "Acme",
    "Address1": "123 Main St",
    "Address2": "",
    "City": "Suite 100",
    "State": "New York",
    "PostalCode": "NY",
    "PK_Country": 10
  },
  "Mailing": {
    "FirstName": "Mary"
  },
  "Phone": [{
    "PK_Provider": 1,
    "PK_Country": 1,
    "Phone": "310-500-1000",
    "Extension": "44",
    "Type": "x",
    "ValidationCode": 12345,
    "Validated": "1"
  }
}
```

```

}, {
  "Phone": "310-500-5000",
  "Extension": "1234",
  "ValidationCode": "1272818",
  "Validated": "0"
}}
}

```

Notes:
Implementation Instructions:
Known issues
Foreign Requests:
Tables:

account	Account
account	Address
account	Account_Phone

Permissions:

edit_account_basic	Can edit basic account information, like name, address
--------------------	--

Module: [account](#) Service: [account](#)

GET [account/x/accounts](#)

/account/account/account/x/accounts

Description:

View the child accounts for account x. This is not included by default with the account information because in the case of dealers this can be a large number and a search would be better.

Arguments:

Sort: optional value 'Desc'. If not it will be Asc by default.

Count: number of accounts returned

Start: number to start in current account iteration

Limit: default 50, max 200

OrderBy: PK_Account, FirstName, LastName, Name (note that Name is the same as FirstName, it is the Address FirstName)

Response:
Success Code

200


```
{
  "Accounts": [
    {
      "PK_Account": "1",
      "FirstName": "John",
      "LastName": "Doe",
      "Address1": "123 Main St",
      "Address2": "Suite 100",
      "City": "New York",
      "State": "NY",
      "PostalCode": "100121",
      "PK_Country": 1,
      "PK_AccountType": 1,
      "PK_Oem_Account": "abcdef"
    },
    {
      "PK_Account": "2",
      ...
    }
  ]
}
```

Notes:

Implementation Instructions:

Known issues

Foreign Requests:

Tables:

account	Account
---------	---------

Permissions:

lookup_account	Can search for accounts as long as they're children of the authenticated account
----------------	--

Module: [account](#) Service: [account](#)

POST [account/x/installation](#)

[/account/account/account/x/installation](#)

Description:

Add an installation to this account

Arguments:

FirstName:

LastName:

Company:

Address1:

Address2:

City:

State:

PostalCode:

PK_Country:

Response: **Success Code** 201

```
{
  "Location": "http://billing5.mios.com/billing/account/18727/installation/927",
  "PK_Installation": 12345
}
```

Notes:
Implementation Instructions:
Known issues
Foreign Requests:
Tables:

account	Installation
---------	--------------

Permissions:

edit_account_basic	Can edit basic account information, like name, address
--------------------	--

Module: [account](#) Service: [account](#)

DELETE [account/x/installation/y](#)

[/account/account/account/x/installation/y](#)

Description:

Delete the installation y from account x

Arguments:

Response: Success Code 200

returns a 403 Forbidden if this is the only installation on the account

returns a 404 Not Found if the installation is invalid

Notes:**Implementation Instructions:**

Do not let the user delete the only remaining installation. Pick one of the installations and update Device set EK_Installation=[this one] where EK_Installation=Y to move all installations to an existing one

Known issues**Foreign Requests:****Tables:**

account	Installation
---------	--------------

Permissions:

edit_account_basic	Can edit basic account information, like name, address
--------------------	--

Module: [account](#) Service: [account](#)

GET [account/x/devices](#)

/account/account/account/x/devices

Description:

View the devices for account x

Arguments:

Response: **Success Code** 200

```
{
  "Devices": [{
    "PK_Device": 1,
    "PK_DeviceType": 2,
    "PK_DeviceSubType": 3,
    "MacAddress": "aa:bb:cc:dd:ee:ff",
    "Server_Device": "device1.mios.com",
    "Server_Device_Alt": "device2.mios.com",
    "PK_Installation": 15,
```

```
"Name": "John's device",
"Using_2G": 0,
"DeviceAssigned": "12:15:39 01/10/2015",
"Msisdn": "46734962806",
"Imei": "990001321724460",
"Platform": "Vera"
}}
}
```

Notes:
Implementation Instructions:
Known issues
Foreign Requests:
Tables:

auth_device	Device
account	Device

Permissions:

view_account	Can view basic account information (address, phone, etc.), but not billing
--------------	--

Module: [account](#) Service: [account](#)

PUT account/x/device/y

/account/account/account/x/device/y

Description:

Update information on this device (PK_Installation is the only option now)

Arguments:

PK_Installation: Update to this installation index

Response: **Success Code** 200

Notes:
Implementation Instructions:
Known issues
Foreign Requests:

Tables:

account	Device
---------	--------

Permissions:

edit_account_basic	Can edit basic account information, like name, address
--------------------	--

Module: [account](#) Service: [account](#)

GET [account/x/users](#)

/account/account/account/x/users

Description:

View the users for account x.

Arguments:

Response:	Success Code	200		
<pre>{ "Users": [{ "PK_User": 1, "PK_Account": 1, "PK_PermissionRole": 10, "Username": "testuser", "Name": "name", "Email": "test@mailinator.com", "ValidationCode": 12345678, "Validated": 0, "EmailAlerts": 1, "EmailNotifications": 1, "PhoneNotifications": 1, "PushNotifications": 1, "Devices": "1,2,3,4,5", "Created": "2014-06-24 07:04:03", "PK_Oem_User": "IDFORM-320232-pjYyDKJLqZssKeZ6+6666699UT8CjkJHA", "Phone": [{ "PK_Provider": -1, "PK_Country": 1, "Phone": "12345678", </pre>				

```

        "Extension": "",
        "ValidationCode": "",
        "Validated": 1
    }
]
}
]
}

```

Notes:

Implementation Instructions:

Known issues

Foreign Requests:

Tables:

account	User
---------	------

Permissions:

view_account	Can view basic account information (address, phone, etc.), but not billing
--------------	--

Module: [account](#) Service: [account](#)

GET account/x/installations

/account/account/account/x/installations

Description:

View the installations for account x.

Arguments:

Response: **Success Code** 200

```

{
  "Installations": [{
    "PK_Installation": 11,
    "FirstName": "John",
    "LastName": "Smith",
    "Address1": "Somewhere 12",
    "Address2": "",
    "City": "New York",
    "PostalCode": "12345",

```

```
"PK_Country": 2,
"Company": "Cucumber",
"Msisdn": "46734962806",
"Iimei": "990001321724460",
"Devices": [{
  "PK_Device": 1
}, {
  "PK_Device": 44
}]
}
```

Notes:
Implementation Instructions:
Known issues
Foreign Requests:
Tables:

account	Address
account	Installation

Permissions:

view_account	Can view basic account information (address, phone, etc.), but not billing
--------------	--

Module: [account](#) Service: [account](#)

DELETE [account/x/user/y](#)

[/account/account/account/x/user/y](#)

Description:

Delete user y from account x

Arguments:

Response: **Success Code** 200

Notes:
Implementation Instructions:

Known issues**Foreign Requests:****Tables:**

account	User
---------	------

Permissions:

edit_account_basic	Can edit basic account information, like name, address
--------------------	--

Module: [account](#) Service: [account](#)**GET [account/x/installation/y](#)**[/account/account/account/x/installation/y](#)**Description:**

Return data for installation y for account x

Arguments:**Response:** **Success Code** 200

```
{
  "PK_Installation": 11,
  "FirstName": "John",
  "LastName": "Smith",
  "Address1": "Somewhere 12",
  "Address2": "",
  "City": "New York",
  "PostalCode": "12345",
  "PK_Country": 2,
  "Company": "Cucumber",
  "Msisdn": "46734962806",
  "Imei": "990001321724460",
  "DeviceAssign": "2014-06-24 07:04:03",
  "ServicePlan": "disable",
  "ServicePlanState": "state"
}
```

Notes:**Implementation Instructions:****Known issues**

Foreign Requests:**Tables:**

account	Address
account	Installation

Permissions:

view_account	Can view basic account information (address, phone, etc.), but not billing
--------------	--

Module: [account](#) Service: [account](#)**PUT [account/x/installation/y](#)**[/account/account/account/x/installation/y](#)**Description:**

Update installation address

Arguments:

FirstName:

LastName:

Company:

Address1:

Address2:

City:

State:

PostalCode:

PK_Country:

Response: **Success Code** 200**Notes:****Implementation Instructions:****Known issues****Foreign Requests:****Tables:**

account	Address
---------	---------

account	Installation
---------	--------------

Permissions:

edit_account_basic	Can edit basic account information, like name, address
--------------------	--

Module: [account](#) Service: [account](#)

GET account/x/accounts/count

/account/account/account/x/accounts/count

Description:

View the child accounts count for account x.

Arguments:

Response:	Success Code	200
<pre>{ "ChildrenCount": 1 }</pre>		

Notes:
Implementation Instructions:
Known issues
Foreign Requests:
Tables:
Permissions:

Module: [account](#) Service: [account](#)

PUT account/x/device/y/name

/account/account/account/x/device/y/name

Description:

Update "Name" information on this device (PK_Installation is the only option now)

Arguments:

Name: new name for the device

Response:	Success Code	200
------------------	---------------------	-----

Notes:**Implementation Instructions:****Known issues****Foreign Requests:****Tables:****Permissions:**

set_device_status	Can set device status
-------------------	-----------------------

Module: [account](#) **Service:** [installation](#)**Module Description**

Handles account information and personal information for accounts, users, such as names and addresses.

Service Description

Requests related to installations

Module: [account](#) **Service:** [installation](#)**PUT installation/x/attribute**

/account/installation/installation/x/attribute

Description:

Change installation attributes

ServicePlan=disable - will disable the gateway tokens for gateway on the installation x. Any other value will enable it back.

Arguments:

[Msisdn]

[ServicePlan]

[ServicePlanState]

Response:**Success Code**

200

Notes:**Implementation Instructions:**

Known issues**Foreign Requests:****Tables:**

account	Installation
device	Sync

Permissions:

add_installation_attribute	Can add installation attributes
----------------------------	---------------------------------

Module: [account](#) **Service:** [User](#)**Module Description**

Handles account information and personal information for accounts, users, such as names and addresses.

Service Description

Requests related to managing users

Module: [account](#) Service: [user](#)

PUT user/x

/account/user/user/x

Description:

Update PK_User=x information

Arguments:

```
{
  "PK_Account": 1,
  "PK_PermissionRole": 10,
  "Username": "johnlogin",
  "Name": "Smith",
  "Password": "currentpasswordifdifferentuserontheaccount",
  "Email": "john@somewhere.com",
  "EmailAlerts": 1,
  "EmailNews": 0,
  "EmailNotifications": 1,
  "PhoneNotifications": 1,
  "PushNotifications": 1,
  "NotifyOnSeverity": [1, 2, 3, 4],
  "Phone": {
    "PK_Phone": "112",
    "Type": "",
    "PK_Provider": 1,
    "PK_Country": 2,
    "Phone": "111-222-333",
    "Extension": "111",
    "ValidationCode": 1234,
    "Validated": 0
  }
}
```

Response: **Success Code** 200

Notes:

Implementation Instructions:

Known issues

Foreign Requests:

Tables:

account	Address
account	User
account	Phone

Permissions:

alter_user	Can alter user info. All users have this, although only some have this permission for child users
------------	---

Module: **account** Service: **user**

GET user/x

/account/user/user/x

Description:

Get the details for PK_User=x

Arguments:

Response:	Success Code
{	200

```
{
  "PK_Account": 1,
  "PK_PermissionRole": 10,
  "Username": "johnlogin",
  "Name": "Smith",
  "Email": "john@somewhere.com",
  "ValidationCode": 1234,
  "Validated": 0,
  "EmailAlerts": 1,
  "EmailNews": 0,
  "EmailNotifications": 1,
  "PhoneNotifications": 1,
  "PushNotifications": 1,
  "NotifyOnSeverity": [1, 2, 3, 4],
  "Devices": "1111,2222,3333,4444",
  "Phone": {
    "PK_Phone": "112",
    "Type": "",
    "PK_Provider": 1,
    "PK_Country": 2,
    "Phone": "111-222-333",
    "Extension": "111",
    "ValidationCode": 1234,
    "Validated": 0
  }
}
```

Notes:

Implementation Instructions:

Known issues

Foreign Requests:

Tables:

account	Account
account	User
account	Phone

Permissions:

view_user	Can view user info. All users have this, although only some have this permission for child users
-----------	--

Module: **autha** Service: **auth**

Module Description

Authenticate accounts and users

Service Description

Create and authenticate users and manage passwords

Module: **autha** Service: **auth**

GET username/x

/autha/auth/username/x

Description:

Authenticate user with the username x.

Arguments:

Url Parameters

username: Url-encoded and lowercase version of the username.

TokenVersion

PortalHash

SHA1Password: Concatenate the username+password+oZ7QE6LcJp6fiWzdqZc - and generate a SHA1 key. NOTE: username must be lowercase.

before concatenation.

PK_Oem: The ID of the OEM

[PK_Account_Parent]: The account ID of the parent (dealer, distributor, etc.)

AppKey: key assigned to app by master account

Response:	Success Code	200
------------------	---------------------	-----

Example response:

{
 "Identity":
 "eyJFeHBpcmVzIjoxMzkzMzE3NjUzLCJHZW5lcmF0ZWQiOjEzOTcyMzEyNTMsIlBlcm1pc3Npb25zIjpbey
QJS19QZXJtaXNzaW9ulfoxLcJNB2RlljoxfSx7IIBLX1Blcm1pc3Npb24iOiJsIk1vZGUiOiJF9LHsiUEtfUGVybyWlzc2lvbi6MywiTW9kZSI6MX0seyJQS19QZXJtaXNzaW9uljo0LcJNB2RlljoxfSx7IIBLX1Blcm1pc3Npb24iOiYjslk1vZGUiOiJF9LHsiUEtfUGVybyWlzc2lvbi6OSwiTW9kZSI6MX0seyJQS19QZXJtaXNzaW9uljoxNCwiTW9kZSI6MX0seyJQS19QZXJtaXNzaW9uljoxNSwiTW9kZSI6MX0seyJQS19QZXJtaXNzaW9uljoxNywiT
W9kZSI6MX0seyJQS19QZXJtaXNzaW9uljoxOCwiTW9kZSI6MX0seyJQS19QZXJtaXNzaW9uljoxOSwiT
W9kZSI6MX0seyJQS19QZXJtaXNzaW9uljoyMCwiTW9kZSI6MX0seyJQS19QZXJtaXNzaW9uljoyMSwiT
W9kZSI6MX0seyJQS19QZXJtaXNzaW9uljoyMiwiTW9kZSI6MX0seyJQS19QZXJtaXNzaW9uljoyMywiT
W9kZSI6MX0seyJQS19QZXJtaXNzaW9uljoyNSwiTW9kZSI6MX0seyJQS19QZXJtaXNzaW9uljoyNiwiT
W9kZSI6MX0seyJQS19QZXJtaXNzaW9uljoyNywiTW9kZSI6MX0seyJQS19QZXJtaXNzaW9uljoyOCwiT
W9kZSI6MX0seyJQS19QZXJtaXNzaW9uljoyOSwiTW9kZSI6MX0seyJQS19QZXJtaXNzaW9uljozMCwiT
W9kZSI6MX0seyJQS19QZXJtaXNzaW9uljo0MiwiTW9kZSI6MX0seyJQS19QZXJtaXNzaW9uljo1MiwiTW
9kZSI6MX0seyJQS19QZXJtaXNzaW9uljo2MiwiTW9kZSI6MX0seyJQS19QZXJtaXNzaW9uljo3MiwiTW9
kZSI6MX0seyJQS19QZXJtaXNzaW9uljo4MiwiTW9kZSI6MX0seyJQS19QZXJtaXNzaW9uljo5MiwiTW9k
ZSI6MX0seyJQS19QZXJtaXNzaW9uljoxMDIsIk1vZGUiOiJF9LHsiUEtfUGVybyWlzc2lvbi6MTyLcJNB2Rllj
oxfSx7IIBLX1Blcm1pc3Npb24iOiEYmiWiTW9kZSI6MX0seyJQS19QZXJtaXNzaW9uljoxMzIsIk1vZGUiOiJF
9LHsiUEtfUGVybyWlzc2lvbi6MTQyLcJNB2RlljoxfSx7IIBLX1Blcm1pc3Npb24iOiE1MiwiTW9kZSI6MX0sey
JQS19QZXJtaXNzaW9uljoxNjIsIk1vZGUiOiJF9LHsiUEtfUGVybyWlzc2lvbi6MTcyLcJNB2RlljoxfSx7IIBLX1B
lcm1pc3Npb24iOiE4MiwiTW9kZSI6MX0seyJQS19QZXJtaXNzaW9uljoxOTIsIk1vZGUiOiJF9LHsiUEtfUGV
ybWlzc2lvbi6MjAyLcJNB2RlljoxfSx7IIBLX1Blcm1pc3Npb24iOiIxMiwiTW9kZSI6MX0seyJQS19QZXJtaX
NzaW9uljoyMjIsIk1vZGUiOiJF9LHsiUEtfUGVybyWlzc2lvbi6MjMyLcJNB2RlljoxfSx7IIBLX1Blcm1pc3Npb24i
Ojl0MiwiTW9kZSI6MX0seyJQS19QZXJtaXNzaW9uljoyNTIsIk1vZGUiOiJF9LHsiUEtfUGVybyWlzc2lvbi6MjYyLcJNB2RlljoxfSx7IIBLX1Blcm1pc3Npb24iOiJ3MiwiTW9kZSI6MX0seyJQS19QZXJtaXNzaW9uljoyODIsIk1vZGUiOiJF9LHsiUEtfUGVybyWlzc2lvbi6MjkycLcJNB2RlljoxfSx7IIBLX1Blcm1pc3Npb24iOiJMwMiwiTW9kZSI6MX0seyJQS19QZXJtaXNzaW9uljozMTIsIk1vZGUiOiJF9LHsiUEtfUGVybyWlzc2lvbi6MzlyLcJNB2RlljoxfSx7IIBLX1Blcm1pc3Npb24iOiMzMzMiwiTW9kZSI6MX0sevJQS19QZXJtaXNzaW9uljozNDIsIk1vZGUiOiF9

}


```
LHsiUEtfUGVyBwLzc2lvbil6NDgyLCJNb2RlIjoxfSx7IIBLX1Blcm1pc3Npb24iOjQ5MiwiTW9kZSI6MX0seyJ
QS19QZXJtaXNzaW9uljo1ODIsIk1vZGUiojF9LHsiUEtfUGVyBwLzc2lvbil6Njg1LCJNb2RlIjoxfSx7IIBLX1B
lcm1pc3Npb24iOjcwNSwiTW9kZSI6MX0seyJQS19QZXJtaXNzaW9uljoxNTk2LCJNb2RlIjoxfSx7IIBLX1B
lcm1pc3Npb24iOjE2MDYsIk1vZGUiojF9LHsiUEtfUGVyBwLzc2lvbil6MTYxNiwiTW9kZSI6MX0seyJQS19Q
ZXJtaXNzaW9uljoxNjI2LCJNb2RlIjoxfV0sIIBLX0FjY291bnQiOjEwOTk4ODE3LCJQS19BY2NvdW50VHlw
ZSI6NSwiUEtfQWNjb3VudENoaWxkljowLCJQS19BY2NvdW50X1BhcmVudCI6NTIyLCJQS19PZW0iOjE
sIIBLX1Blcm1pc3Npb25Sb2xlljoxMCwiUEtfVXNlciI6MTE5OTc4NDcsIIBLX1NlcnZlci9BdXRoljoxLCJQS19
TZXJ2ZXJfQWNjb3VudCI6NiwiUEtfU2VydMvYX0V2ZW50ljoXNCwiU2VydMvYX0F1dGgiOiJtbXMTZGV2
MTEubWlvcy5jb20iLCJVC2VybmFtZSI6Im1pb3NkZW1vMSJ9",
  "IdentitySignature":
    "qT7fnpuFHaohPY9Dg4ANF7JtTaf+S0oS/7vqjieOuHKt69evlwQPP5Uy5Baj1qUzaSfFHHQvxlZArbX4LzCu
    xwquqkPzdRoWIVX74jxkZ0V/DsS8uTRvzp9+CrSX3N8jOqGFPxVOnsWTA/rQb0icyEExWys1BqRS5dAp
    XwijLG1NMqBQHjAPIHWRmMgtwCaBPlz9hFCW+eLwxASs/BMphsGRvpp8JUgXrCiwzSoH6+FkjpJlhmT
    mG/oruSZz0U4N+SR6WnBdJHAuQRik6e5w3sanYMG5N4lG3haGdPpx3uJvRfE9cUrSAUkgQSQ==",
  "Server_Event": "mms-dev24.mios.com",
  "Server_Event_Alt": "mms-dev14.mios.com",
  "Server_Account": "mms-dev24.mios.com",
  "Server_Account_Alt": "mms-dev14.mios.com"
}
```

Identity: base64 encoded string, which after decoding in a json string with following items:

PK_User: The user id

PK_Account: The account id

PK_AccountType: The account type

PK_PermissionRole: The permission role

Expires: unix timestamp

Generated: unix timestamp

Username: name of user for which the Identity is signed (same as parameter)

PK_Server_Auth: The ID of auth server

PK_Oem_User: id of the user in the external oem system, if any (Uid)

List of items ends here

The following parameters are on the same level as Identity:

IdentitySignature: binary string encoded as base64, signature corresponding to the Identity

Server_Event: The main event server

Server_Event_Alt: The backup event server

Server_Account: The main account server

Server_Account_Alt: The backup account server

Login to other servers with this identity is done by passing Identity and IdentitySignature in http headers.

For example in curl:

```
curl -v -k --header "MMSAuth:Identity" --header "MMSAuthSig:IdentitySignature"
```

```
https://sta2-new.mios.com/info/session/token
```

Will get the 128+24 bit session key (32+6 characters, hex encoded) corresponding to the whole Identity (user x here). Then the session key can be saved and used to optimize further access:

```
curl -v -k --header "MMSSession:12345611223344556677889900112233445566"  
https://mms-dev14.mios.com/account/user/user/1
```

If session or identity+signature headers are in the request, servers can return:

403: Forbidden, if permissions are insufficient (request can be retried with other permissions successfully)

401: Unauthorized, if signature check failed, or session is invalid. This means that new session and/or signature should be requested from auth server. Additional error information is in string format:

MMSAuth error="error text". Currently it covers:

"invalid session" -when session expired or simply doesn't exist, new session should be requested

"invalid signature" -signature check failed (retrying the request will always fail). Can be caused by bugs like data corruption, missing part of signature/identity string, wrong encoding (recommended way to use identity and signature is to not reencode them, send the same data that is received from auth server). Identity can be decoded to get the data from it, but instead of reencoding it, save the original, base64 encoded string for purposes of authentication.

"expired token" -signature is valid, but expiration date embedded in identity is too old. New token should be requested from auth server

"invalid token" -signature is valid and not expired, but is missing mandatory data. This should never happen, unless we have a bug in creating tokens on auth server.

Notes:

Implementation Instructions:

Lookup the user/account by the username where the account's OEM matches, and, if parent is specified, any of the FK_Account_Parent fields match.

Mobile apps should send the raw password in SHA1Password in case of SSO.

Known issues

Foreign Requests:

Tables:

auth_account	Account
--------------	---------

auth_account	User
--------------	------

Permissions:

Module: **autha** Service: **auth**

POST username

/author/auth/username

Description:

Create a new user either adding one to an existing account, if this session is authenticated and a valid PK_Account is passed, or create a new account

Arguments:

Username: The desired username

Email: The associated email

Password: plaintext password (should not start or end with \t,\r,\n or space)

PK_Oem: The ID of the OEM:

PK_Account_Parent: The account ID of the parent (dealer, distributor, etc.)

Response:	Success Code	201
------------------	---------------------	-----

The response is the same as with username/x GET.

Notes:
Implementation Instructions:
Known issues
Foreign Requests:
Tables:

config	AccountType
auth_account	Account
auth_account	User

Permissions:

add_user	Authorized to add remove or users to the account of the designated PermissionGroup. If no argument is specified, only the default PermissionGroup will be added, or if *, then any possible PermissionGroup for the account type	Permission Groups(s)
----------	--	----------------------

Module: **autha** Service: **auth**

POST account

/autha/auth/account

Description:

Create a new account

Arguments:

PK_AccountType: The PK_AccountType to create a new account (default: end-user)

PK_Oem: The ID of the OEM (constants assigned by MiOS)

PK_Account_Parent: The account ID of the parent (dealer, distributor, etc.)

[PK_Server_Account]: Optionally the account server to use

Response:

Success Code

201

Json with:

PK_Account: The account

Server_Event: The main event server

Server_Event_Alt: The backup event server

Server_Account: The main account server

Server_Account_Alt: The backup account server

Notes:

Implementation Instructions:

Known issues

Foreign Requests:

Tables:

auth_account	Account
--------------	---------

Permissions:

add_account	Can add new accounts
-------------	----------------------

Module: **autha** Service: **auth**

GET account/x

/autha/auth/account/x

Description:

Lookup account x

Arguments:

Response: **Success Code** 200

Session token

JSON:

PK_AccountType: private key for account

FK_Account_Parent: private key for parent account level x

FK_Account_Parent2:

FK_Account_Parent3:

FK_Account_Parent4:

PK_Oem:

Server_Event: The main event server

Server_Event_Alt: The backup event server

Server_Account: The main account server

Server_Account_Alt: The backup account server

Notes:**Implementation Instructions:****Known issues****Foreign Requests:****Tables:**

auth_account	Account
--------------	---------

Permissions:

view_account	Can view basic account information (address, phone, etc.), but not billing
--------------	--

Module: [autha](#) Service: [auth](#)

GET app/x/token

/autha/auth/app/x/token

Description:

Returns user token with permissions limited for the specified app

Arguments:

User token

Response:	Success Code	200
-----------	--------------	-----

JSON with new login token and account servers:

Identity:

IdentitySignature:

Server_Event:

Server_Event_Alt:

Server_Account:

Server_Account_Alt:

Notes:**Implementation Instructions:**

expiration date of token is preserved

permissions are included only if original token includes them too

Known issues**Foreign Requests:****Tables:****Permissions:**Module: [autha](#) Service: [auth](#)

GET user/x/token/refresh

`/autha/auth/user/x/token/refresh`**Description:**

Create a new token from the current token, to update changed data like permissions. Effectively this is the same as logout and login, but without password and changing token expiration

Arguments:

Session/user token

[AppKey]:

[TokenVersion]:

Response:	Success Code	200
-----------	--------------	-----

New token and signature, with expiration set to the same time as original token.

Identity,

IdentitySignature,
Server_Event,
Server_Event_Alt,
Server_Account,
Server_Account_Alt

Notes:**Implementation Instructions:****Known issues****Foreign Requests:****Tables:****Permissions:**

Module: [autha](#) Service: [auth](#)

POST user/x/logout

/autha/auth/user/x/logout

Description:

Executes log out for user X.

Arguments:

Response:	Success Code	200
------------------	---------------------	-----

Notes:**Implementation Instructions:****Known issues****Foreign Requests:****Tables:****Permissions:**

Module: [authd](#) Service: [auth](#)

Module Description

Authenticate devices

Service Description

Authenticate devices

Module: **authd** Service: **auth**

GET device/x/zwavelocale

/authd/auth/device/x/zwavelocale

Description:

Return ZWaveLocale for device x

Arguments:

MMSSession can be used instead

MacAddress

Response: **Success Code** 200

ZWaveLocale

Format:

BEGIN : device

ZWaveLocale=VALUE

END : device

Notes:**Implementation Instructions:****Known issues****Foreign Requests:****Tables:**

auth_device	Device
-------------	--------

Permissions:

Module: **authd** Service: **auth**

GET device/x/servers

/authd/auth/device/x/servers

Description:

Return server list for device x

Arguments:

[DeviceType]
[MacAddress]
[PK_Oem]

Response:

Success Code 200

```
{
  "Servers": [{
    "PK_Device": 2222,
    "PK_Account": 1234,
    "PK_Oem": 1,
    "Server_Device": "device.mios.com",
    "Server_Event": "event.mios.com",
    "Server_Account": "account.mios.com"
  }, {
    "PK_Device": 3333,
    "PK_Account": 1234,
    "PK_Oem": 1,
    "Server_Device": "device.mios.com",
    "Server_Event": "event.mios.com",
    "Server_Account": "account.mios.com"
  }]
}
```

Notes:

Implementation Instructions:

Known issues

Foreign Requests:

Tables:

auth_device	Device
-------------	--------

Permissions:

Module: [authd](#) Service: [auth](#)

GET device/x/permissions

/authd/auth/device/x/permissions

Description:

Retrieve permissions list for device x

Arguments:

Response:	Success Code	200
json array with permissions		

Notes:**Implementation Instructions:****Known issues****Foreign Requests:****Tables:****Permissions:**Module: **authd** Service: **auth**

/authd/auth/device/x/checklogout/y

GET device/x/checklogout/y**Description:**

Return success if a logout was made after Y timestamp.

Arguments:

Y - represents timestamp

Response:	Success Code	200
------------------	---------------------	-----

Notes:**Implementation Instructions:****Known issues****Foreign Requests:****Tables:****Permissions:**Module: **authd** Service: **auth****POST device/x/logout**

/authd/auth/device/x/logout

Description:

Log out a device.

Arguments:

Response:	Success Code	200
------------------	---------------------	-----

Notes:**Implementation Instructions:****Known issues****Foreign Requests:****Tables:****Permissions:**

Module: [authd](#) Service: [auth](#)

GET devices

/authd/auth/devices

Description:

Search devices using parameters to filter the result

Arguments:

[PK_Device]

[Imei]

At least one search parameter must be specified

Response:	Success Code	200
------------------	---------------------	-----

```
{
  "Devices": [
    {
      "PK_Device": 36000111,
      "PK_Account": 5,
      "Imei": "abc"
    },
    {
      "PK_Device": 36000311,
      "Imei": "def"
    }
  ]
}
```

Notes:**Implementation Instructions:**

Known issues**Foreign Requests:****Tables:****Permissions:**

search_devices	Authorized to search for a device by PK_Device and to see their status.
----------------	---

Module: **device** Service: **device**

Module Description

Maintains the status and detailed info on subset of devices (aka gateways, access points, accessories). Also contains the status on local devices (ie Z-Wave nodes, plugins, etc.)

Service Description

Requests related to the status of devices

Module: **device** Service: **device**

GET device/x/zwavelocale

/device/device/device/x/zwavelocale

Description:

Return ZWaveLocale for device x

Arguments:

MacAddress

Response:

raw string with ZWaveLocale data:

BEGIN

ZWaveLocale=string_data

END

Success Code

200

Notes:**Implementation Instructions:**

This is ZWaveLocale from device.Gateway (see also authd/auth/device/x/zwavelocale

Known issues**Foreign Requests:****Tables:**

device	Gateway
--------	---------

Permissions:

Module: [device](#) Service: [device](#)

GET devices_count

/device/device/devices_count

Description:

Return devices count for authenticated user

Arguments:

Response: **Success Code** 200

JSON:

DevicesOnline: total devices online

DevicesOffline: total devices offline

GatewaysOnline:

GatewaysOffline:

SensorsOnline:

SensorsOffline:

CamerasOnline:

CamerasOffline:

LocksOnline:

LocksOffline:

ThermostatsOnline:

ThermostatsOffline:

OtherOnline: devices not in other categories (except for total)

OtherOffline:

Notes:**Implementation Instructions:**

Known issues**Foreign Requests:****Tables:****Permissions:**

Module: **device** Service: **device**

GET device/x/kit

/device/device/device/x/kit

Description:

Return device X kits

Arguments:

Response:	Success Code	200
-----------	--------------	-----

```
{
  "Kits": [
    "PK_Kit": 1,
    "Transaction": 2,
    "Devices": [
      {
        "DeviceType": "1",
        "DeviceFile": "value",
        "ImplFile": "value",
        "RequireMac": "value",
        "Protocol": "value",
        "Category": "value",
        "SubCategory": "value",
        "Manufacturer": "value",
        "Model": "value",
        "Name": "value",
        "ProductID": "value",
        "ProductType": "value",
        "MfrId": "value",
        "PnpID": "value",
        "PK_KitDevice": "value",
        "Qty": "value",
        "MacAddress": "value"
      }
    ],
    "ExtraDevices": [
      {
        "DeviceType": "1",
```

```
"DeviceFile" : "value",
"ImplFile" : "value",
"RequireMac" : "value",
"Protocol" : "value",
"Category" : "value",
"SubCategory" : "value",
"Manufacturer" : "value",
"Model" : "value",
"Name" : "value",
"ProductID" : "value",
"ProductType" : "value",
"MfrId" : "value",
"PnpID" : "value",
"PK_KitDevice" : "value",
```

```
"Qty" : "value",
"MacAddress": "value"
}
]
]
}
```

Notes:**Implementation Instructions:****Known issues****Foreign Requests:****Tables:****Permissions:**

Module: [device](#) Service: [device](#)

GET [device/x/status](#)

[/device/device/device/x/status](#)

Description:

Return device alive date and online status.

Arguments:

Response:	Success Code	200
{		
"PK_Device" : "1",		
"AliveDate" : "date_format",		
"FirmwareVersion" : "version",		

```
"Online" : ""  
}
```

Notes:**Implementation Instructions:****Known issues****Foreign Requests:****Tables:****Permissions:**

Module: eula Service: eula

Module Description

Module providing EULA for applications

Service Description

Fetch/add/update/log EULA for a certain user/oem

Module: eula Service: eula

GET eula

/eula/eula/eula

Description:

Retrieve the latest EULA version for a certain Oem

Arguments:

PK_User

PK_Oem

AppKey

LangKey

Response:	Success Code
	200

```
{  
  "Eula": "http://eula.url",  
  "EulaVersion": "1.2.3",  
  "PK_Eula": 1  
}
```

Notes:**Implementation Instructions:****Known issues**

Foreign Requests:**Tables:****Permissions:**Module: [eula](#) Service: [eula](#)

POST eula

`/eula/eula/eula`**Description:**

Add a new or update an existing EULA

Arguments:

PK_Oem

AppKey

LangKey

Version

Eula

Response:**Success Code**

200

Notes:**Implementation Instructions:****Known issues****Foreign Requests:****Tables:****Permissions:**Module: [eula](#) Service: [eula](#)

PUT eula

`/eula/eula/eula`**Description:**

Update an existing EULA

Arguments:

PK_Oem

AppKey

LangKey

Version

Eula

Response: **Success Code** 200

Notes:

Implementation Instructions:

Known issues

Foreign Requests:

Tables:

Permissions:

Module: [eula](#) Service: [eula](#)

POST [eula/accept](#)

[/eula/eula/eula/accept](#)

Description:

Log EULA as accepted

Arguments:

PK_Oem

PK_Account

PK_Eula

PK_User

Response: **Success Code** 200

Notes:

Implementation Instructions:

Known issues

Foreign Requests:

Tables:

Permissions:

Module: [event](#) Service: [event](#)

Module Description

Events, like when a device is online, a user logs in, a sensor is tripped, etc. for a subset of accounts

Service Description

Requests related to events

Module: [event](#) Service: [event](#)

GET events

/event/event/events

Description:

Get a list of events

Arguments:

Camera=a&Gateway=b&Unread=1&Type=c&Source=d&Format=e&Device=f&Count=g&Start=h&Before=i
&After=j&User=k&Pass=l&Sort=Asc

The arguments are as follows:

Camera= If specified, this is the id of the camera(s). Only alerts from this camera(s) are included. Multiple cameras should be comma separated. Or specify * for all cameras the user has access to.

Gateway= If specified, this is a list of gateway(s). Only alerts from specified gateway(s) are included. Multiple cameras should be comma separated. Or specify * for all gateways the user has access to.

User= See note below.

- **Note:** If neither a Camera nor a Gateway is specified, only alerts for users who are passed for validation are included (such as login events). If cameras and/or gateways are specified, then user alerts as well as Gateway/Camera alerts are included. If you only want Gateway/Camera alerts, add User=0. This will exclude user events.
- So Gateway=1,3&Camera=* will include alerts from gateways #1 and 3, from all cameras, and also include user alerts. Adding User=0 will exclude the user alerts. If User=0 is specified without any gateway or camera then no records are returned. Note that alerts from a gateway/cam will not be included if the gateway/cam is not already paired to the user's account.

Unread= If this is '1' then only alerts that haven't been read yet are included.

Type= Only alerts of the specified type are included. Alert types are listed below:

- **Source=** Only alerts from this source are included. See [source types](#) below.
- **Format=** Only alerts containing files of this format are included (eg mp4, jpg, mjpeg).
- **Device=** If a gateway is specified, only alerts from this device id on the gateway are included.

Count= Specify how many rows to include.

Start= Starting with this row number

Before= / After= A date range as a unix timestamp (i.e. an integer in UTC time). You cannot use local time because some devices, like cameras, don't have a clock/timezone setting, so the times are universal timestamps applied by the server and not local times on the device.

The return data in json is as follows:

PK_Event= A numeric id of the alert .

PK_Device= The id of the gateway that generated the alert, if any.

PK_User= The id of the user that generated the alert, if any.

PK_Notification= If users were sent a notification of the alert (SMS, email) this is the id of the notification attempts.

PK_Account = return events from a different account, if allowed by permissions

DeviceID= If a device on a gateway generated the alert this is the device id.

Timestamp= The date when the alert was received as a unix timestamp

LocalDate= The date according to the gateway, specified in local time as yyyy-mm-dd hh:mm:ss

Read= The date when the alert was read by the user as a unix timestamp

Expiration= The date when the alert will be automatically deleted as a unix timestamp

EventType= The type of alert:

- **1** - Image (a camera sent an image).
- **2** - Video (a camera sent a video clip).
- **3** - Trigger (a user-created trigger was fired).
- **4** - Variable (a device changed state, such as a door was unlocked, a user code entered, a sensor tripped).
- **5** - Login (a user logged into the portal).
- **6** Gateway Connected (a gateway came online).

SourceType= What caused the alert to be generated:

- **1** - User (e.g. User hit 'record' button or otherwise initiated the alert)
- **2** - Timer (an automatic timer generated it)
- **3** - Trigger (a user-created trigger)
- **4** - Variable (some device changed state)

Format= If there's a file associated with the alert, this is the extension. E.g. mpg, jpg, mjpeg.

IP= The IP address from which the alert was sent.

Locked= If 1 the alert will not automatically delete on expiration

FileSize= If there's a file associated with the alert, this is the size of the file.

Code and NewValue= These depend on the type of alert. For Variables 'code' is the name of the variable (state) that changed and value is the new value.

Description= A description for the alert. For triggers this is the user-defined trigger name.

Key= A unique, random value for each alert. When you need to modify an alert you will pass back the key to confirm you are authorized. If a file is attached the key is also a key on a storage server.

Comments= Comments which the user has associated with the alert.

[Severity]= Comma separated list of severe event types. All events are returned by default (severe and others). Severity=1,2,3,4 will return all events with severity 1-4

[Sort]= Asc or Desc. Default sorting order is descending

[PK_StorageMethod]= id of the storage method, if 3rd party storage is used

[Icon]= icon or the

Response:	Success Code	200
------------------	---------------------	-----

This returns a list of archives in JSON format as follows:

```
{
  "Records": [
    {
      "PK_Event": 192817,
      "PK_Device": "0",
      "PK_User": "",
      "PK_Notification": "1827",
      "DeviceID": "0",
      "Timestamp": "18217271",
      "LocalDate": "",
      "Expiration": "18217271",
      "Read": "",
      "EventType": "1",
      "SourceType": "0",
      "Format": "jpg",
      "Server_Storage": "storage1.mios.com",
      "PK_Store": 12345,
    }
  ]
}
```

```

    "IP": "192.168.2.27",
    "Locked": "0",
    "FileSize": "32728",
    "Code": "",
    "NewValue": "",
    "Description": "Camera Image",
    "Key": "1278",
    "Severity": 0,
    "Thumbnail": "",
    "UserComments": ""
  }
]
}

```

Notes:
Implementation Instructions:
Known issues
Foreign Requests:
Tables:

event	Event
-------	-------

Permissions:

view_events	Can view the events for an account
-------------	------------------------------------

Module: [event](#) Service: [event](#)

GET event/x

/event/event/event/x

Description:

Fetch details on an event

Arguments:

key (event.Event.Key)

Response: **Success Code** 200

This returns details about the alert, including any notification attempts, as follows. Note the status codes for contacts and attempts are in the defined constants section:

```
{
  "PK_Event": 192890,
  "PK_User": 2,
  "Contacts": [
    {
      "PK_Contact": 10,
      "Type": 1,
      "Destination": "john@gmail.com",
      "NumAttempts": 0,
      "Status": "",
      "Contents": "200820:\r\n=\r\nserial:-\r\n",
      "PK_User": 4,
      "Attempts": [
        {
          "Timestamp": 182821717,
          "Status": "1",
          "Notes": "busy"
        },
        {
          "Timestamp": 1812717,
          "Status": "2",
          "Notes": "complete"
        }
      ]
    }
  ],
  {
    "PK_Contact": 20,
    "Type": 3,
    "Destination": "+1310-555-0800",
    "NumAttempts": 0,
    "Status": "",
    "Contents": "200820:\r\n=\r\nserial:-\r\n",
    "PK_User": 6
  }
]
```

Notes:

Implementation Instructions:

Known issues

Foreign Requests:

Tables:

event	Event
-------	-------

Permissions:

view_events	Can view the events for an account
-------------	------------------------------------

Module: [event](#) Service: [event](#)

PUT event/x

/event/event/event/x

Description:

Alter the data for an event x, marking it read, locked, deleted, and adding/modifying comments
x can be used to alter multiple events, when it is in comma separated format

User needs permission "read_events" to mark events as read. "delete_events" is required to delete, lock or unlock an event and add comments to it

Arguments:

Read: [0|1]

Lock: [0|1]

Delete: [0|1]

Comments: Some user comments on the alert

[PK_Account]: account id for which the attributes should be applied (default is the current token account)

Response: **Success Code** 200

Notes:
Implementation Instructions:
Known issues
Foreign Requests:
Tables:

event	Event
-------	-------

Permissions:

read_events	Can mark events as read
delete_events	Can delete, lock and unlock events.

Module: **event** Service: **event**

GET data_usage

/event/event/data_usage

Description:

Returns data usage for authenticated user

Arguments:

Response: **Success Code** 200

JSON:

DataUsage: current usage in bytes (approximated)

Allowance: available space for user in bytes (approximated)

Note that numbers returned are strings, in case of problems with json parsers supporting bigger than 32-bit values

Notes:

Implementation Instructions:

Known issues

Foreign Requests:

Tables:

Permissions:

Module: **firmware** Service: **build**

Module Description

Manage firmware builds

Service Description

Requests related to builds that have been completed on the build server

Module: **firmware** Service: **build**

PUT firmwareupgrade

/firmware/build/firmwareupgrade

Description:

Executes a firmware upgrade for a list of PK_Device.

Arguments:

PK_Device
PK_Oem
PK_Account_Parent
ForceUpgradeUrl
ForceUpgradeKeepSettings
ForceUpgradeVersion

Response:	Success Code	200
Servers updated list in a json format.		

Notes:**Implementation Instructions:**

Users must have the UPDATE_FIRMWARE permission.

Known issues**Foreign Requests:****Tables:****Permissions:**

Module: **firmware** Service: **firmware**

Module Description

Manage firmware builds

Service Description

Requests related to firmware

Module: **firmware** Service: **firmware**

/firmware/firmware/firmware

GET firmware

Description:

Retrieve latest version of the firmware

Arguments:

Mandatory:

PK_Oem

Optional:

PK_Language

Platform

UI

UISkin

BetaKey

Response: **Success Code** 200**Single item:**

```
{"Version":"1.7.1142","Url":"http://builder1204.mios.com/mt7620a_betafirmware/mt7620a_Luup_ui7-1.7.1142-en-mios.squashfs","Type":"","Platform":"","Language":"","UI":"","UISkin":""}
```

Multiple items:

```
[{"Version":"1.6.157","Url":"http://download.mios.com/rt3662_betafirmware/rt3662_Luup_ui6-1.6.157-plperaverde.squashfs","Type":"squashfs","Platform":"SercommNA900","Language":"6","UI":"6","UISkin":"peraverde"},
```

```
{"Version":"1.6.157","Url":"http://download.mios.com/rt3662_betafirmware/rt3662_Luup_ui6-1.6.157-plperaverde.squashfs","Type":"squashfs","Platform":"MiCasaVerde VeraLite","Language":"6","UI":"6","UISkin":"peraverde"}]
```

Notes:**Implementation Instructions:****Known issues****Foreign Requests:****Tables:****Permissions:**Module: [firmware](#) Service: [firmware](#)

POST firmware

[/firmware/firmware/firmware](#)**Description:**

Add a firmware version and url

Arguments:

Mandatory:

Version

Url

PK_Oem

Optional:
Platform
UI
UISkin
PK_Language

Response: **Success Code** 200

Notes:
Implementation Instructions:
Known issues
Foreign Requests:
Tables:
Permissions:

Module: [firmware](#) Service: [firmware](#)

PUT firmware

/firmware/firmware/firmware

Description:
Updates a firmware data

Arguments:
Mandatory:
Version
Url
PK_Oem
Optional:
VersionFrom
VersionTo
UI
UISkin

Response: **Success Code** 200

Notes:
Implementation Instructions:
Known issues
Foreign Requests:
Tables:
Permissions:

Module: [info](#) Service: [config](#)

Module Description

This module is always available and contains general information about the running MMS instance.

Service Description

Server static configuration

Module: [info](#) Service: [config](#)

POST refresh

/info/config/refresh

Description:

Refresh static data cache

Arguments:

Response: Success Code

Notes:

Implementation Instructions:

Known issues

Foreign Requests:

Tables:

Permissions:

Module: [info](#) Service: [config](#)

GET config

/info/config/config

Description:

Return server static configuration data.

This should be cached by the mms client for at least the duration of the user session.

Arguments:

Configuration category has to be added as GET parameter:

AccountType=1

Response:

json with static data

Success Code

200

AccountType
Country
Country_Provider
Currency
DeviceCategory
DeviceType
DeviceSubType
FileType
Item
Language
PaymentMethod
Permission
PermissionCategory
PermissionRole
PermissionRole_Permission
Server
Server_Module
StorageType
Terms

Notes:**Implementation Instructions:****Known issues****Foreign Requests:****Tables:****Permissions:**

Module: [info](#) Service: [session](#)

Module Description

This module is always available and contains general information about the running MMS instance.

Service Description

Requests related to session handling

Module: [info](#) Service: [session](#)

GET token

/info/session/token

Description:

Return new session id for this server.

Arguments:

In http headers:

MMSAuth: X

MMSAuthSig: Y

X is the base64 encoded identity information

Y is the binary signature, base64 encoded

Proper values are received from [GET autha/auth/username/x](#) or [GET authd/auth/device/x](#)

Response:	Success Code	200
------------------	---------------------	-----

38(32+6) byte session id hex encoded string

Session id can be used in custom http header MMSession in format:

MMSession:0000011234123412341234123412341234

Notes:

Implementation Instructions:

Known issues

Foreign Requests:

Tables:

Permissions:

Module: [info](#) Service: [session](#)

GET session/x

/info/session/session/x

Description:

Retrieves session Identity details from Session id.

Arguments:

Previous session id in http headers:

MMSession=XXXXXX

Response:	Success Code	200
------------------	---------------------	-----

Session Id info in JSON format:

Notes:

Implementation Instructions:

Known issues

Foreign Requests:

Tables:

Permissions:

Module: locator Service: locator

Module Description

Locates devices

Service Description

Locate devices

Module: locator Service: locator

GET locator

/locator/locator/locator

Description:

Returns a list of devices detected on the same network as the caller

Arguments:

[ExternallIP]: searches for devices with this ExternallIP only. When ExternallIP is not specified, the request will use the current IP of the incoming connection

[PK_Account]:

[LastAliveSeconds]: only gateways that are alive in last x seconds (default days=864000)

Response:	Success Code
	200

```
{
  "Devices": [{
    "PK_Device": 12345,
    "MacAddress": "aa:bb:cc:dd:ee:ff",
    "PK_DeviceType": 1,
    "PK_DeviceSubType": 2,
    "Server_Device": "device.mios.com",
    "Server_Event": "event.mios.com",
    "PK_Account": 10,
    "Server_Account": "account.mios.com",
    "InternallIP": "1.2.3.4",
    "LastAliveReported": "2016-01-02 10:55:19",
    "Platform": "Vera"
  }]
}
```

Notes:

Implementation Instructions:

Known issues

Foreign Requests:

Tables:

auth_device	Device
-------------	--------

Permissions:

Module: [locator](#) Service: [locator](#)

GET device/x/servers

/locator/locator/device/x/servers

Description:

Retrieves device, account and event servers for device X.

Arguments:

DeviceType -optional

Response:	Success Code
{ "PK_Account" : 1, "Server_Device" : "server.address", "Server_Event" : "server.address", "Server_Account" : "server.address" }	200

Notes:**Implementation Instructions:****Known issues****Foreign Requests:****Tables:****Permissions:**

Module: [locator](#) Service: [locator](#)

GET macaddress/x/servers

/locator/locator/macaddress/x/servers

Description:

Retrieves device, account and event servers for Mac address X.

Arguments:

DeviceType -optional

Response:	Success Code	200
{		
"PK_Device" : 1,		
"PK_Account" : 1,		
"Server_Device" : "server.address",		
"Server_Event" : "server.address",		
"Server_Account" : "server.address"		
}		

Notes:**Implementation Instructions:****Known issues****Foreign Requests:****Tables:****Permissions:**Module: [locator](#) Service: [locator](#)

GET oem/x/servers

/locator/locator/oem/x/servers

Description:

Retrieves server device for Oem X.

Arguments:

Response:	Success Code	200
{		
"Servers":[
{		
"PK_Server_Device" : "1",		
"Server_Device" : "server.address"		
},		
{		
"PK_Server_Device" : "1",		
"Server_Device" : "server.address"		
}		
]		

```
}
```

Notes:**Implementation Instructions:****Known issues****Foreign Requests:****Tables:****Permissions:**

Module: [relay](#) Service: [relay](#)

Module Description

The SSH tunneling for remote access

Service Description

Relay to/from gateways

Module: [relay](#) Service: [relay](#)

GET device/x/usage

/relay/relay/device/x/usage

Description:

Get the usage statistics for device x

Arguments:

From

To

Response:**Success Code**

200

Json:

```
{"Usage": "12345"}
```

Notes:**Implementation Instructions:****Known issues****Foreign Requests:****Tables:****Permissions:**

Module: [relay](#) Service: [relay](#)

POST relay/device/x

/relay/relay/relay/device/x

Description:

Sending a HTTP request to gateway x using the control tunnel
request POST /relay/relay/device/x/abc/def will be changed to POST /abc/def
request GET /relay/relay/device/x/abc/def will be changed to GET /abc/def

Arguments:

[MMSSession]: optional

[Method]: [PUT/GET/POST/DELETE] -override the http method in relayed request

Response: **Success Code** 200

Responds with tunnel response or error

Notes:**Implementation Instructions:****Known issues****Foreign Requests:****Tables:**

relay	SshTunnel
-------	-----------

Permissions:

Module: relay Service: relay

POST uiredirect/device/x

/relay/relay/uiredirect/device/x

Description:

Redirect request to UI

Arguments:

RedirectUrl: relative address on relay server, for example www/1.2.3?a=b&c=d

Host: optional

MMSAuth

Response: **Success Code** 200

Notes:

Implementation Instructions:

GET/POST

set cookie with auth data and send redirect response: MiOS=a,b,c,d

where:

a -auth token

b -auth token signature

c -Server_Account

d -Server_Account_Alt

Known issues**Foreign Requests:****Tables:****Permissions:**Module: [relay](#) Service: [relay](#)

PUT account/x/limitusage

/relay/relay/account/x/limitusage

Description:

Update usage limits for Account X.

Arguments:

ResetCameraCount

ResetRaCount

RaLimitKb

CameraLimitKb

Response:	Success Code	200
------------------	---------------------	-----

Notes:**Implementation Instructions:****Known issues****Foreign Requests:****Tables:****Permissions:**Module: [relay](#) Service: [relay](#)

GET account/x/limitusage

/relay/relay/account/x/limitusage

Description:

Retrieve usage limits for Account X.

Arguments:**Response:** **Success Code** 200

JSON data:

```
{  
  "RaDateReset" : 100,  
  "CameraDateReset" : 100,  
  "RaKb" : 100,  
  "CameraKb" : 100,  
  "RaLimitKb" : 100,  
  "CameraLimitKb" : 100  
}
```

Notes:**Implementation Instructions:****Known issues****Foreign Requests:****Tables:****Permissions:**

Module: [storage](#) Service: [storage](#)

Module Description

Store large data, such as logs, camera images, backup files, video clips

Service Description

Requests to store and retrieve large binary data

Module: [storage](#) Service: [storage](#)

POST store

/storage/storage/store

Description:

Store a binary object on the server for later retrieval. Used for logs, backup files, camera images and video, etc.

Arguments:

PK_Account:

PK_Device:

PK_User:

Format:

Filesize:

Md5:

Version:

Tags:

PK_StorageType:

[HttpHeader]: 1 if file includes http header (i.e. Mjpeg)

[Filename]: original filename (if it needs to be retrieved later)

Data:

Response: **Success Code** 201

PK_Store: The ID of the item

Key: a 32 bit key to retrieve this

Notes:**Implementation Instructions:**

Updates storage.StoreStatistics

Known issues**Foreign Requests:****Tables:**

storage	Store
---------	-------

Permissions:Module: [storage](#) Service: [storage](#)**GET store/x**

/storage/storage/store/x

Description:

Get the storage item x information

Arguments:

Key: The key to retrieve the item
output_format for raw data

Response:**Success Code**

200

Format: text or Json

PK_Store:

PK_Account:

PK_Device:

PK_User:

Format:

Version:

Tags:

Filesize:

Md5:

Expiration:

Filename:

PK_StorageType:

Timestamp:

Notes:**Implementation Instructions:****Known issues****Foreign Requests:****Tables:**

storage	Store
---------	-------

Permissions:Module: [storage](#) Service: [storage](#)**GET store/x/archive**

/storage/storage/store/x/archive

Description:

Return stored file

Arguments:

Key: 32-bit key received from POST store/x

RawMjpeg: optional if equals "1" the file will be output in raw format, otherwise attachment

Response: file data (binary) **Success Code** 200

Notes:

Implementation Instructions:

HttpHeader=1 will enable workaround for mjpeg streaming

Known issues

Foreign Requests:

Tables:

storage	Store
---------	-------

Permissions:

Module: [storage](#) Service: [storage](#)

GET store/x/thumbnail

/storage/storage/store/x/thumbnail

Description:

Get thumbnail for storage item x

Arguments:

Key: 32-bit key created during POST store/x
x: PK_Store of previously stored item

Response: binary thumbnail data **Success Code** 200

Notes:

Implementation Instructions:

currently thumbnail is created on demand

Known issues

Foreign Requests:

Tables:

Permissions:

Module: [storage](#) Service: [storage](#)

GET store

/storage/storage/store

Description:

Returns storage items filtered by provided parameters

Arguments:

[Tags]:

[Format]:

[Version]:

[Filesize]:

[Md5]:

[Filename]:

[PK_StorageType]:

[Sort]: Asc or Desc, default order is descending

[Start]: starting item number, default 0

[Count]: number of items to retrieve, default 10

Response:**Success Code**

200

Based on optional param output_format: text or JSON

PK_Store:

PK_Account:

PK_Device:

PK_User:

Format:

Version:

Tags:

Filesize:

Md5:

Expiration:

Filename:

PK_StorageType:

Timestamp:

Notes:**Implementation Instructions:****Known issues****Foreign Requests:****Tables:****Permissions:**

Module: [storage](#) Service: [storage](#)

DELETE store/x

/storage/storage/store/x

Description:

Delete the storage item x and related information

Arguments:

Key: The key to retrieve the item

Response:	Success Code	200
------------------	---------------------	-----

Notes:**Implementation Instructions:****Known issues****Foreign Requests:****Tables:****Permissions:**

Module: [techsupport](#) Service: [techsupport](#)

Module Description

Everything related to tech support

Service Description

Requests to manage trouble tickets

Module: [techsupport](#) Service: [techsupport](#)

POST device/x

/techsupport/techsupport/device/x

Description:

Add Device X energy consumption.

Arguments:

Json:

```
{
  "Date" : "date time format"
  "Data" : [
    {
      "DeviceID" : "11",
      "Avg" : "11",
      "Peak" : "123"
    }
  ]
}
```

Response:

Success Code

201

Notes:

Implementation Instructions:

Known issues

Foreign Requests:

Tables:

Permissions: